Signalogic DSP Software

System Guide

Signalogic, Inc.
Dallas, Texas

Rev. 2
Sept, 1997

The software described in this document is furnished under license agreement. The software may be used or copied only under the terms of this license agreement (see section 3.1.1).

Copyright © 1994-1997 Signalogic, Inc. All rights reserved.

No part of this manual may be photocopied or reproduced in any form without prior written consent from Signalogic, Inc.

U.S. Government: If Licensee is acquiring the software on behalf of any unit or agency of the United States Government, the following shall apply:

(a) for units of the Department of Defense:
RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data Clause at DFARS 252.227-7013.

(b) for any other unit or agency:
NOTICE -- Notwithstanding any other lease or license agreement that may pertain to, or accompany the delivery of, the computer software and accompanying documentation, the rights of the Government regarding use, reproduction and disclosure are as set forth in Clause 52.227-19(c)(2) of the FAR.

Contractor/manufacturer is Signalogic, Inc., 9617 Wendell @Skillman, Dallas, TX 75243.

Hyperception is a registered trademark of Hyperception, Inc.
Signalogic and DSPower are registered trademarks, and Real-Time Convolver is a trademark of Signalogic, Inc.
MATLAB is a registered trademark of The MathWorks, Inc.
National Instruments is a registered trademark, and NIDSP is a trademark of National Instruments, Inc.
Borland is a registered trademark of Borland International, Inc.
Microsoft is a registered trademark, and MS-DOS, Windows, and Visual C are trademarks of Microsoft Corporation.
Analog Devices, AT&T, Motorola, NEC, and Texas Instruments are registered trademarks of their respective holders. ADSP-21xx, WE DSP32x, DSP5600x, DSP9600x, 77220, and TMS320xx are trademarks of their respective holders.
Crystal Semiconductor is a registered trademark of Crystal Semiconductor, Inc.
PC is a trademark of IBM Corporation.
All other trademarks belong to their respective holders.

Signalogic, Inc.
9617 Wendell @ Skillman
Dallas, TX 75243
214-343-0069

Signalogic DSP Software System Guide, Sept1997
Design and Implementation: Jeff Brower.

Programming: Jeff Brower and Varsha Shamabhat.
Documentation: Jeff Brower, Varsha Shamabhat, Hanh Nguyen.

Contents

1 Essential Information **1**

Overview 1
1.1 Relationship of Signalogic DSP Software Components 2
1.2 Summary of Signalogic DSP Software Components 10
1.2.1 Hypersignal®-Macro Series Software Summary 10
1.2.2 DSPower-HWLib Software Summary 87
1.2.3 DSP Source Code Interface Software Summary 87
1.2.4 DSPower-Block Diagram Software Summary 91
1.2.5 DSPower-C/C++ Code Generator Software Summary 92
1.2.6 Real-Time Convolver Software Summary 92
1.3 System Requirements 93
1.4 System Considerations 102
1.4.1 Memory Considerations 103
1.4.2 Performance Considerations 106
1.4.3 Video and Display Considerations 112
1.4.4 System Drivers and Resident Program Considerations 115
1.4.5 DSP/Analog Hardware Considerations 116
1.5 Technical Support 117

2 Documentation **115**

Overview 115
2.1 Documentation Assumption of Previous Knowledge 116
2.2 Documentation Organization 117
2.3 Documentation Conventions 119
2.4 Documentation References and Additional Sources 124

3 Installation **127**

Overview 127

3.1 Before You Install 128
3.1.1 Signalogic DSP Software License Agreement 128
3.1.2 Media and Disk Space Requirements 132
3.1.3 Documentation and Media You Should Have Received 135
3.2 Installing Signalogic DSP Software 138
3.2.3 Configuration and Checkout of DSPower-HWLib Software 143
3.2.4 Configuration and Checkout of DSPower-Block Diagram Software 147
3.2.5 Installing and Configuring Hypersignal Software For User Mode Operation 149
3.3 Release Notes 175
3.3.1 Revision History for Hypersignal-Macro Series Software 176
3.3.2 Revision History for DSPower-HWLib Software 225
3.3.3 Revision History for DSPower-SPLib Software 226
3.3.4 Revision History for DSP Source Code Interface Software 227
3.3.5 Revision History for DSPower-Block Diagram Software 229
3.4 Directory Structure Chart 230
3.5 Uninstalling 232

4 Troubleshooting 231

Overview 231
4.1 Troubleshooting Installation Problems 232
4.2 Troubleshooting Video and Display Problems 236
4.3 Troubleshooting DSP/Analog Hardware Problems 237
4.3.1 I/O-Mapped Hardware Conflicts 238
4.3.2 Memory-Mapped Transfer Gotchas 239
4.3.3 Host PC Internal Bus Speed Conflicts 248
4.3.4 Data Acquisition Problems 249

Glossary of Terms 253

Index 257

Glossary of Terms

.ide

Filename extension used for Borland C/C++ v4.5 project files.

.mak

Filename extension used for Microsoft Visual C/C++ v1.5 project files.

.prj

Filename extension used for Borland C/C++ v3.1 project files.

API

Application Programming Interface. Refers to interface presented by DLL or other Windows object to external applications.

DDE

Dynamic Data Exchange. A Windows 3.x compatible method of transferring data between applications which don't "know" about each other. Likely to be replaced by the faster and more reliable OLE method under Windows '95.

DLL

Dynamic Link Library. A library of routines which can be linked to any number of applications on demand, at run-time. The library is treated as an object, to which applications know how to interface, but not how it functions internally.

DRAM

Dynamic random access memory. Typically used as large amount of auxiliary memory; for a DSP/analog board, but sometimes used instead of SRAM if the DSP device has enough onchip SRAM. 4-5 times slower and less expensive than SRAM.

EX

Extended. Used to refer to extended functions or applications. Examples include the f_fftex and f_adex functions, and the Hypersignal-Macro EX software.

expanded memory

Additional, usually large, amount of memory added to a PC system, but accessed through a small (64 kbyte) segment in conventional memory space. No longer as prevalent with advent of systems configured with 4 Mbyte or more of standard (extended) memory.

extended memory

System memory located above 1024 kbyte. This is the memory referred to when a system is described as having 4 Mbyte or RAM, or 16 Mbyte of RAM, etc. installed.

IDE

Integrated Development Environment. Refers to marketing term used by popular C/C++ compiler manufacturers to describe a code development software package where all functions can be performed from the same, graphical user-interface.

ISR

Interrupt Service Routine. Used frequently in discussing the structure and function of DSP source code.

MLS

Maximum Length Sequence. A pseudo-random noise sequence of digital 1s and 0s mathematically defined to not repeat for N-1 samples, where N= 2**^{M}, and M is the sequence order.

NDA

Non-Disclosure Agreement.

NRT

Non-Real-Time. Commonly used to refer to non-real-time versions of software. Hypersignal-Macro NRT is an example.

RT

Real-Time. Commonly used to refer to real-time versions of software.

OLE

Object Linking and Embedding. Refers to a Windows compatible method for applications which otherwise don't "know" each other to share information and data.

SRAM

Static random access memory. Typically used as main program and data memory for a DSP/analog board. 4-5 times faster and more expensive than DRAM.

VxD

Virtual Device Driver. A type of Windows device driver used to perform low-level and/or priveleged system functions or communication.

Index

Error! No index entries found.

1 Essential Information

Overview

If you take the time to read anything in Signalogic DSP Software documentation, read the Essential Information section. All subjects are presented from a system-level perspective, with the idea that they might be read before the software is used, or at least while it is still being configured. A "development hierarchy" of Signalogic DSP Software explaining software component relationships is presented first, followed by a summary of individual software package capabilities. Other important sections include System Requirements, Performance Considerations, and DSP/Analog Hardware Considerations.

Main Topics

Relationship of Signalogic DSP Software Components
Hypersignal®-Macro Series Software Summary
DSPower®-HWLib Software Summary
DSP Source Code Interface Software Summary
DSPower-Block Diagram Software Summary
DSPower-C/C++ Code Generator Software Summary
Real-Time Convolver™ Summary
System Requirements
Memory Considerations
Performance Considerations
Video and Display Considerations
System Drivers and Resident Program Considerations
DSP/Analog Hardware Considerations
Technical Support

1.1 Relationship of Signalogic DSP Software Components

There are three main relationships between Signalogic DSP Software components:

1) Functional superset or subset, where there exists a version of a component that is a superset or subset of another in some way. Examples include real-time capability, and specific application capability.

2) DSP Engine, a hierarchical relationship where one component can make programmatic and transparent use of another, and overall functionality and convenience is available to the user by dealing only with the user-interface of the controlling component.

3) Source code generation capability, where one or more types of source code generation is not included in the standard version of a component, for marketing and cost reasons.

Functional Supersets and Subsets

Currently, functional supersets and subsets in Signalogic DSP Software components apply only to Hypersignal-Macro series software. These relationships are explained below:

1) Hypersignal-Macro Non-Real-Time. This package is a subset of Hypersignal-Macro. It contains the same simulation, file acquisition, waveform file displays, and DSP and math functions as Hypersignal-Macro, but has no real-time capability. Instruments, function acceleration using the DSP, anything else hardware-related is not active.

2) Hypersignal-Macro EX is a superset of Hypersignal-Macro. It is different only in that it contains additional macro language programs and functions.

3) Hypersignal-Acoustic is a superset of Hypersignal-Macro EX, containing three levels of increased functionality:

- additional macro language programs and functions
- additional intrinsic functionality inside displays and instruments
- additional device drivers for specialized acoustic/audio hardware

4) Nihon versions of Hypersignal packages. These versions are configured to run in DOS mode on NEC PC98xx series machines commonly found in Japan.

For more information, see section 1.2.1, Hypersignal-Macro Series Software Summary.

Related Topics

Functional Supersets and Subsets

DSP Engines

DSPower and DSP Engines

User Mode and Engine Mode

Programs, Template Files, and Function Files

Pipe.386

DSP Engines

Mainly due to the critical need for testability in increasingly complex systems, Signalogic's DSP software has evolved in a hierarchical relationship, with mature, reliable, off-the-shelf packages playing an important role.

The idea of using existing, off-the-shelf software packages in their

object form--used as they are documented except controlled programmatically--has led to development of the "DSP Engine" concept. General application of this concept allows source code to be generated for not only Signalogic off-the-shelf software packages, but also other software packages, such as MATLAB® 4.x for Windows, and supported DSP/analog boards. Theoretically, the only requirement is that the engine be able to accept and run either source or object versions of programs. Of course, practical application is determined by considerations such as the type and popularity of language used, availability of development tools, etc.

Use of the DSP Engine concept also makes the overall system easier for users to understand and debug. Also, upgrading from one level to another is straightforward and cost-effective, since previous packages are still useful. And most importantly, the concept makes expansion of system capabilities simpler and more reliably implemented and tested for Signalogic. See also DSPower and DSP Engines.

Related Topics

Functional Supersets and Subsets

DSP Engines

DSPower and DSP Engines

User Mode and Engine Mode

DSPower and DSP Engines

[@j external application, controlling application, external controlling application]

The DSPower series of packages embodies a division-of-labor concept that relies on separate, well-defined "DSP Engines" to perform basic DSP, math, and hardware tasks. DSP Engines are expected to be off-the-shelf packages; in some cases, they may alternatively be able to run stand-alone, with their own user-interface (user mode operation).

In the DSPower-Block Diagram environment, user-interface and source code generation are performed by DSPower, while at least one or more DSP engines and various support tools are invoked remotely and transparently to achieve compilation and execution of generated code, as well as interactive operation. Signalogic believes that this represents a fundamental design advantage over other packages that perform simulations internally, and then generate source code as a separate process--which can produce inconsistent results and debug-resistant problems. Several system simplifications also result from the DSPower architecture; for example, block-diagrams are saved and retrieved in source code form. Currently there are three (3) DSP

engine possibilities: MATLAB (version 4.x for Windows, available from The MathWorks), Hypersignal-Macro, Hypersignal-Macro EX, or Hypersignal-Acoustic (version 4.x for DOS, available from Hyperception or Signalogic), and a supported PC-compatible DSP/analog board from one of several manufacturers. Both MATLAB and Hypersignal-Macro offer superb simulation, by virtue of their rich procedural languages containing many functions and displays, and built-in compilers.

For example, the DSPower-Block Diagram package can instruct Hypersignal package (v4.0 or higher), MATLAB (v4.0 or higher), or a DSP/analog hardware platform to do useful work, under remote control, and transparently to the user. DSP Engines are responsible for the actual implementation of DSP/math functions, displays / visualizations / instruments, digital filter design, waveform file manipulation, etc. In the case of a supported DSP/analog hardware platform, it may be possible to execute complete block diagrams by downloading the compiled and linked form of DSPower-generated C code.

As another example, in the DSPower-HWLib DLL and DSPower-FileLib architectures, these DLLs communicates with a VxD (virtual device driver), which communicates directly with hardware drivers, waveform file

functions, and DSP/math functions located in a DSP Engine. Engine examples include accessing the hardware drivers inside a Hypersignal-Macro series package (real-time versions of which include drivers for over 60 types of DSP/analog hardware), or accessing DSP/math functions inside MATLAB from a C/C++ program. DSP Engine communication is achieved both by creating native DSP Engine programming language source code (e.g. Hypersignal macro language source, or .m file source), by sending programming via the VxD, or direct Windows messaging.

Related Topics

Functional Supersets and Subsets

DSP Engines

DSPower and DSP Engines

User Mode and Engine Mode

Programs, Template Files, and Function Files

Pipe.386

User Mode and Engine Mode

[@j simulated keyboard, user mode, engine mode]

Software applications that can be used as DSP Engines by DSPower-Block Diagram can operate in one of two *operating modes: user mode* or *engine mode*. In user mode, the application has been invoked manually, and is under control of an operator. In engine mode, the software application has

been invoked programmatically, and is under the control of an outside Windows application (external controlling application; see DSP Engines, and DSPower and DSP Engines, above). In engine mode, visual aspects of the engine's user-interface are typically suppressed, and input is supplied to the engine in two main methods:

1. Through generation of an initial program instructing the engine. To do this the controlling application generates source code in the native language for the engine, compiles it (or submits it to the engine for compiling), and gives it to the engine for execution. Examples include Hypersignal macro language and MATLAB .m files.

2. By supplying input to the engine as if coming from a "simulated keyboard", which allows operation of the engine to be modified and controlled dynamically, in complement to, or as a replacement for, the initial program given to the engine.

Normally, a description of input required to control a DSP Engine function will be accurate for both user mode and engine mode, except the input must be presented to the DSP Engine in a different manner. For example, in the case of Hypersignal-Macro series software packages, the macro language in these packages has a simulated keyboard input facility, which can accept input from the actual keyboard, the macro language program (interactive commands; see Hypersignal-Macro Series Users Manual section 7.3.2, Issuing Interactive Commands From Macros), or, when Hypersignal is operating in engine mode, from an external application using the Pipe.386 VxD. This makes it possible for external applications to

control all aspects of Hypersignal-Macro series software operation using debugged, well-documented input commands.

In some cases, though, certain types of input in user mode may not apply in engine mode. This can occur because some part of the user interface, or some visual cue, is not available in engine mode, or more often, because in engine mode the input is taken care of automatically. When this occurs in decriptions of Hypersignal-Macro series software in Signalogic DSP Software documentation, the section or heading is marked clearly with "In User Mode", to highlight the expectation that the section only pertains if the software is being operated in user mode.

Related Topics

Functional Supersets and Subsets

DSP Engines

DSPower and DSP Engines

User Mode and Engine Mode

Programs, Template Files, and Function Files

Pipe.386

Programs, Template Files, and Function Files

[@j program files, template files, function files, reduced architecture]

The DSP Engine concept naturally leads to a division of native language source code generated by DSPower series software and

submitted to DSP Engines into two types:

1. programs
2. functions

At run-time, programs are the text file equivalent of an onscreen block or block diagram, representing some DSP, math, display, instrument, etc. operation, or combination of operations, to be performed. Programs make one or more calls to functions. Because there are many blocks in the DSPower system, and potentially an unlimited number of block diagrams, the actual code content of programs must be kept to a minimum.

In the DSPower-Block Diagram architecture, "template files" are used as the basis for creating source code representing ready-to-execute DSP Engine programs, which then call "function files". Much of the content of template files is non-source code information: block registration, connection definitions, parameter descriptions, graphical control interface descriptions, current onscreen status, etc.

Functions comprise the bulk of "working code" in the DSPower architecture. They are not typically altered by DSPower series software, which means they can exist in the system in either source or executable form. A given function might be invoked by many different programs. DSPower-Block Diagram software maintains libraries of function files which

represent the basic capabilities of the system, and tend not to overlap.

One major benefit of the program / function subdivision is to allow tasks being executed from either user mode or engine mode to share the bulk of the source code needed (function files), to reduce the possiblility of error or differences in results. The same function files are being used, although the user mode program which does the calling may look much different than the DSPower template file. See User Mode and Engine mode, above.

Another benefit of the program / function subdivision is to reduce the actual amount of source code in the overall system by creating a fundamental, orthogonal set of function files that can be called in various combinations or sequences to do just about any task of which DSP Engines are capable.

The overall result of the DSPower architecture is a system both easy for the user to understand, and for Signalogic to maintain.

Related Topics

Functional Supersets and Subsets

DSP Engines

DSPower and DSP Engines

User Mode and Engine Mode

Programs, Template Files, and Function Files

Pipe.386

Pipe.386

The Pipe.386 VxD (virtual device driver) is included with DSPower series software components, allowing them to communicate with DOS DSP Engines such as Hypersignal-Macro and Hypersignal-Acoustic.

The VxD concept in Windows is somewhat of a "magic key", allowing system communication and operation that is not normally documented by Microsoft or otherwise obvious. In the case of DSPower, Pipe.386 is used to send interactive macro language instructions and receive processing and display results from the macro language "simulated keyboard" entry facility in Hypersignal packages. Use of a VxD to perform this communication means that it is ultrafast, or at least as fast as in possible in Windows, and not subject to protocol and other restrictions of a method such as DDE.

Use of a VxD allows one or more Hypersignal packages to be "active" and performing tasks designated by a DSPower series software package, yet not visible from the DSPower user-interface.

Related Topics

Functional Supersets and Subsets

DSP Engines

DSPower and DSP Engines

User Mode and Engine Mode

1.2 Summary of Signalogic DSP Software Components

Summaries of major Signalogic DSP Software components are presented to help answer questions such as "what can it do?" and "which package do I need?". These sections also help explain the relationship between packages. See Appendix A for a development hierarchy chart that helps explain the relationship between Signalogic DSP Software components, and other off-the-shelf packages with which they are compatible.

1.2.1 Hypersignal®-Macro Series Software Summary

The Hypersignal-Macro series of software packages includes Hypersignal-Macro NRT, Hypersignal-Macro, Hypersignal-Macro EX, and Hypersignal-Acoustic. There are also "Nihon" versions of these packages modified to run on NEC PC98xx-compatible machines.

Hypersignal-Macro is a DOS program with menu-driven and programming language user interfaces; it offers math/DSP

simulation, instrumentation, filter design, data acquisition, and display and interactive analysis. Hypersignal-Macro can also operate as a "DSP Engine" with DSPower, in which case it is controlled transparently by DSPower. Hypersignal-Macro supports more than 60 types of DSP/analog, acquisition, and multimedia hardware.

Hypersignal-Macro EX is a superset of Hypersignal-Macro which includes additional functions such as long FFT, convolution, and correlation, frequency zoom, utility macros, and more.

Hypersignal-Macro NRT is a "non-real-time" version of Hypersignal-Macro, which offers the same simulation and file-based processing function, but no real-time support. The package contains no device drivers for DSP/acquisition hardware; all input data is expected to be imported using the built-in File Acquisition function, which is capable of gathering data from a wide variety of binary and ASCII file formats, with control over data precision, header filtering, multichannel, complex, and other format characteristics. Hypersignal-Macro NRT is typically available for 1/2 the cost of Hypersignal-Macro.

Hypersignal-Acoustic is a superset of Hypersignal-Macro EX, and includes additional functions such as MLS stimulus & response, equalization, Schroeder integration, minimum phase calculation,

reverse waterfall, additional Spectrum Analyzer and waveform display capability, device drivers for TEF20HI, DSP-16, ProPort/DatPort, and Tahiti/Monterey.

NOTE: In the software function overviews below, "Hypersignal software" is sometimes used as shorthand to mean "Hypersignal-Macro series software".

Below is a list of functions, displays, instruments, and utilities inside Hypersignal series software packages, as well as system consideration topics.

Hypersignal Functions

DSP Simulation Function Overview

FFT Generation

Convolution/Correlation

LPC Autocorrelation

Recursive Filtering

Power Spectra Generation

Transfer Function Generation

Inverse FFT Generation

Difference Equations

Sampling Rate Conversion

Hypersignal-Macro EX Functions

Long FFT

Long Convolution/Correlation

Frequency Zoom

Minimum Phase

Wavelet Transform

Hypersignal-Acoustic Functions

Schroeder Integration

Equalization

MLS Stimulus & Response

Hypersignal Digital Filter Design

Digital Filter Design Overview

FIR Filter Design

IIR Filter Design

Filter DSP Code Generation

Pole-Zero Display

File-Based Filter Simulation

"Snap-In" Real-Time Filtering

Hypersignal Interactive Display

Interactive Display Overview

Time Domain Waveform Display/Edit

Frequency Domain Magnitude and Phase Display

Joint Time-Frequency Waterfall Display

Joint Time-Frequency Contour Display

Filter Design Display

Pole-Zero Display

Hypersignal Instrumentation & Measurement

Instrumentation & Measurement Overview

Spectrum Analyzer

Digital Oscilloscope

Continuous Record/Output To/From Disk

Stimulus & Response

Signal Synthesis

Hypersignal Macro Language

Macro Language Overview

Macro Record Mode

Macro Programs: Ready-to-Execute Functions

Creating Macro Programs and Functions

Macro Language Variables, Constructs, and Procedures

Hypersignal File Acquisition/Conversion Functions

File Import/Export Functions

Tim2Wav and Wav2Tim Waveform File Conversion

Hypersignal External Program Interface

External Program Control

Hypersignal System Considerations

User-Interface

Online Help

Graphics and Mouse Support

80x87 Math Coprocessor Support

Printer Support

Documentation

Hypersignal DSP Simulation Function Overview

Hypersignal-Macro series software DSP simulation functions accept input in time or frequency domain waveform file format, producing time or frequency domain output waveform files. Input time domain file formats can be Hypersignal .TIM or Microsoft .WAV. Output time domain file formats are in Hypersignal .TIM format for most functions; additional Hypersignal-Macro EX functions allow direct .WAV format output. A tim2wav format conversion function is included with all Hypersignal packages. All frequency domain output waveform files are in Hypersignal formats. (For more information about waveform file formats, see Signalogic DSP Software Reference Guide sections 4.1 and 4.2, Hypersignal Waveform File Format, and Microsoft .WAV Waveform File Format).

FFT Generation in Hypersignal

[@k transform,, Fast Fourier]

All Hypersignal-Macro series software packages contain a built-in complex input Fast Fourier Transform (FFT) function. In addition, when operating in user mode, the Hypersignal-Macro EX and Hypersignal-Acoustic packages offer a lfft.mac program which allows a "long FFT" function to be invoked from the Macro Manager menu.

Both the built-in FFT function and the f_fft.mac long FFT function (which is called by the lfft.mac program), can be called from outside Hypersignal, when operating in engine mode.

The built-in FFT function can accept input waveform, framesize, FFT length, window (Hamming, Hanning, Blackman, Bartlett, Gaussian, rectangular, user-defined), and overlap as input parameters. Output magnitude and phase waveform data are generated always; output real and imaginary data can optionally be generated for complex frequency domain operations. For the built-in FFT function, actual FFT processing may be performed either by the host PC or by DSP hardware currently selected as a "DSP Accelerator".

Input waveform size is not limited, and can be segmented according to framesize and overlap selections.

Current maximum FFT length is 4096 pt. for the built-in version (may be less for some DSP hardware). Current maximum length for the lfft.mac long FFT function, is 262,144, although it may be less depending on available host PC system memory (see Signalogic DSP Software System Guide section 1.3, System Requirements).

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Convolution/Correlation in Hypersignal

[@k circular correlation, crosscorrelation, autocorrelation]

All Hypersignal-Macro series software packages contain built-in convolution, cross-correlation, circular correlation, and autocorrelation functions. In addition, when operating in user mode, the Hypersignal-Macro EX and Hypersignal-Acoustic packages offer lconv.mac and lcorr.mac programs which allow "long convolution" and "long correlation" functions to be invoked from the Macro Manager menu.

Both the built-in Convolution/Correlation function and the f_conv.mac long function (which is called by the lconv.mac and lcorr.mac programs), can be called from outside Hypersignal, when operating in engine mode.

Both the built-in functions and long versions can accept input waveform, filter waveform, and output waveform as input parameters. Filter waveforms can be the result of standard Hypersignal functions such as FIR and IIR Filter Construction and Spectrum Analyzer transfer function storage, or any arbitrary time domain waveform. For the built-in Convolution/Correlation function, actual processing may be performed either by the host PC or

by DSP hardware currently selected as a "DSP Accelerator".

Current maximum filter waveform size is 16384 pt. (single-channel data), and 8192 pt. (two-channel data) for the built-in functions. Current maximum filter lengths for the lconv.mac and lcorr.mac long functions is about 180,000 pts. for single-channel data, although it may be less depending on available host PC system memory (see Signalogic DSP Software System Guide section 1.3, System Requirements, for system RAM requirements with Hypersignal-Macro EX). Input waveform size is not limited.

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

LPC Autocorrelation in Hypersignal

[@k autocorrelation,, LPC, linear predictive coding]

All Hypersignal-Macro series software packages contain a built-in LPC (Linear Predictive Coding) Autocorrelation function.

LPC autocorrelation is often used on periodic or semiperiodic data to construct a series of filters with frequency responses that represent the major frequency domain energy areas in the input data. The input data is segmented into frames, and a filter is constructed on each frame of input data, to allow for time-varying characteristics in the input data. The resulting series of "analysis filter" coefficients can be used to create a series of impulse responses, which, when strung together, represent a "facsimile" of the input data. A form of compression is inherent in this process, because only the filter coefficients are needed to represent the original input data, with the overall compression ratio being the input framesize divided by the number of coefficients. And, the

resulting series of frequency response frames, resulting from either the filter coefficients directly or the synthesized data is "smooth", showing only the major energy areas.

LPC autocorrelation function input paramters include input waveform, framesize, filter length, window (Hamming, Hanning, Blackman, Bartlett, Gaussian, rectangular, user-defined), and output waveform. Either analysis or reflection filter coefficients may be chosen. Maximum filter length is 31; input waveform size is not limited.

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Recursive Filtering in Hypersignal

[@k filtering,, IIR, filtering,, recursive]

All Hypersignal-Macro series software packages contain a built-in recursive filtering function, which can apply IIR filters in factored or direct form to input time domain waveform data.

Recursive filtering function input parameters include input waveform, filter coefficient file, and output waveform. Input filter coefficient files are in ASCII format and can be standard output from the Hypersignal Filter Construction functions, or user-defined. The coefficient file format accepts both factored and direct-form numerator (poles) and denominator (zeros), and can be applied in "biquad cascade" (factored) form or direct form.

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Power Spectra Generation in Hypersignal

[@k spectra estimation]

All Hypersignal-Macro series software packages contain a built-in frequency domain function which estimates the power spectra from a magnitude waveform created by previous FFT Generation.

Power Spectra Generation input parameters include input waveform filename, method of estimation, and output waveform. The method

of estimation can be "group averaging", where the input waveform is segmented into one or more sections which are averaged together independently, creating that number of output frames, "running averaging", where a specified number of frames from the input waveform are averaged together, producing a new output frame each time the "boxcar" increments one frame, and "exponential averaging", where a one-pole lowpass filter (with selectable time constant) is applied between input frames.

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Transfer Function Generation in Hypersignal

[@k deconvolution]

All Hypersignal-Macro series software packages contain a built-in frequency domain function which calculates the complex ratio of two waveforms presumably representing the output and input of an "unknown system".

Input parameters include input and output waveform filenames. Output magnitude, phase, and complex waveforms are produced.. The complex output can be used to calculate the equivalent time domain impulse response.

Transfer function analysis is often used to characterize unknown systems when both the input and output of the system are unknown (as opposed to stimulus & response measurement, when the input is known; see below). Assuming that the spectral content of the input was wideband, or was wideband over a period of time due to averaging, the magnitude result can represent the frequency response of the unknown system

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Inverse FFT Generation in Hypersignal

[@k FFT,, inverse, transform,, Fast Fourier]

All Hypersignal-Macro series software packages contain a built-in complex input inverse Fast

Fourier Transform (iFFT) function. In addition, when operating in user mode, the Hypersignal-Macro EX and Hypersignal-Acoustic packages offer a lifft.mac program which allows a "long iFFT" function to be invoked from the Macro Manager menu.

Both the built-in iFFT function and the f_ifft.mac long iFFT function (which is called by the lifft.mac program), can be called from outside Hypersignal, when operating in engine mode.

The built-in iFFT function can accept input waveform filename as an input parameter. Output is a time domain waveform. For the built-in iFFT function, actual iFFT processing may be performed either by the host PC or by DSP hardware currently selected as a "DSP Accelerator".

Input waveform size is not limited.

Current maximum FFT length is 4096 pt. for the built-in version (may be less for some DSP hardware). Current maximum length for the f_ifft.mac long iFFT function, is 262,144, although it may be less depending on available host PC system memory (see Signalogic DSP Software System Guide section 1.3, System Requirements).

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Difference Equations in Hypersignal

[@k equations,, difference]

All Hypersignal-Macro series software packages contain a Difference Equations function which allows generation of new waveforms and manipulation of existing waveforms using various operators and math functions, specified in classic difference equation form. Input sources include existing waveforms, unit-

step and delta functions, and previous output (recursive) data. Operators include signal arithmetic, multiplication, division (and integer division) modulus, differentiation, integration, min and max, and a frequency domain "frame operator". Functions include several trigonometric and transcendental functions, random noise, and various linear and non-linear math functions.

With the Hypersignal Difference Equations, it is straightforward to build sine, pulse, noise, etc. function generators, as well as to operate arbitrarily on existing waveforms. Equations are based on a discrete time index "n", which begins at zero and increments a specified number of times, according to user entry. The difference equation evaluates at each discrete time index. Note that it is also possible to write expressions for n, both when used as an input index, and also when used as an output index, on the left side of the equals sign. Several example equations are shown below. Consider the following equation, which calculates a running average:

$$y[n] = y[n-1] + x1[n] - x1[n-Fr]$$

The equation below would generate a sine waveform:

$$y[n] = \sin(2*pi*n*Fc/Fs)$$

where "Fs" refers to the desired sampling frequency, and "Fc" to the desired sine wave frequency. This equation can be generalized to

a function generator that describes one period of the desired equation (by replacing "F(n)" below with the appropriate expression), and then replicates the initial period as many times as necessary:

$$y[n] = F(n)*(u[n] - u[n-M]) + y[n-M]$$

In the above equation, M specifies the period in samples. A simple example would be to replace F(n) with n, which would create a sawtooth generator. The equation below would "spectrally invert" a time domain waveform, causing dc to be exchanged for half the sampling rate:

$$y[n] = x1[n]*(-1)^n$$

Finally, consider multiplying a time domain waveform by a complex exponential to produce an in-phase real component and a quadrature imaginary component, where Fc=cutoff frequency, and Fs=sampling frequency:

$$y[n] = x1[n]*\cos[2*pi*Fc*n/Fs] - j*x1[n]*\sin[2*pi*Fc*n/Fs]$$

When sample manipulation, by itself or in combination with signal mathematics, is required, the method is equally simple and elegant. The equation below would delay a waveform by "M" samples:

$$y[n] = x1[n-M]$$

The equation below treats x1 as a circular buffer, moving samples an amount specified by M, where Len is the length of x1 and "%" indicates modulus division:

$$y[n] = x1[(n+M)\%Len]$$

Additional Difference Equations Topics

Input Sources in Difference Equations

Arithmetic Operators in Difference Equations

Non-Arithmetic Operators in Difference Equations

Constants in Diffference Equations

Trig, Transcendental, and Calculus Functions in Difference Equations

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Input Sources in Difference Equations

[@k difference equations input sources]

Below is a partial list of possible input sources ("exp" represents the discrete time index, as explained in the main discussion of Difference Equations, and can be an expression of the sample index "n" or the frame index "m"):

Table 1 List of Difference Equations Input Sources

		refers to waveform specified by x1 INPUT FILENAME field;
		refers to waveform specified by x2 INPUT FILENAME field;
		refers to previous output, sometimes called recursive input or feedback
		unit-step function; accesses to this array return 0 if exp < 0 and

	1 if exp ≥>= 1
	delta function; accesses to this array return 0 if exp ≠<> 0 and 1 if exp = 0

Additional Difference Equations Topics

Input Sources in Difference Equations

Arithmetic Operators in Difference Equations

Non-Arithmetic Operators in Difference Equations

Constants in Diffference Equations

Trig, Transcendental, and Calculus Functions in Difference Equations

Arithmetic Operators in Difference Equations

[@k operators,, difference equations, difference equations operators]

Table 2 Difference Equations Arithmetic Operators

Additional Difference Equations Topics

Input Sources in Difference Equations

Arithmetic Operators in Difference Equations

Non-Arithmetic Operators in Difference Equations

Constants in Diffference Equations

Trig, Transcendental, and Calculus Functions in Difference Equations

Non-Arithmetic Operators in Difference Equations

[@k operators,, difference equations, difference equations operators]

Table 3 Difference Equations Non-Arithmetic Operators

	The "<" and ">" characters are MIN and MAX operators, respectively
	The frame operator. A comma causes operands to be read as "frame, offset"

Additional Difference Equations Topics

Input Sources in Difference Equations

Arithmetic Operators in Difference Equations

Non-Arithmetic Operators in Difference Equations

Constants in Diffference Equations

Trig, Transcendental, and Calculus Functions in Difference Equations

Constants in Difference Equations

[@k difference equations constants]

Table 4 Difference Equations Constants

	returns πpi, where πpi = 3.14159265...
	returns value of SAMPLING FREQUENCY field
	returns value of FRAMESIZE field
	sqrt(−1)√−1

Additional Difference Equations Topics

Input Sources in Difference Equations

Arithmetic Operators in Difference Equations

Non-Arithmetic Operators in Difference Equations

Constants in Diffference Equations

Trig, Transcendental, and Calculus Functions in Difference Equations

Trig, Transcendental, and Calculus Functions in Difference Equations

[@k function,, difference equations, difference equations functions]

Table 5 Difference Equations Functions

	|arg|
	returns, in

		radians, arccosine of arg
		returns, in radians, arcsine of arg
		returns, in radians, arctangent of arg
		cosine of arg; arg expected in radians
		cotangent of arg; arg expected in radians

		returns derivative of array[exp] value
		*e***arge^{arg}, where *e* = 2.71828182..
		returns fractional part of real value
		returns integral of array[exp] value

		removes fractional part of real value
		log10(arg)$\log_{10}$(arg)
		ln(arg)
		returns pseudo-random real number between 0 and arg, inclusive
		rounds arg to nearest integer
		rounds arg to

	nearest integer
	returns –1 for arg < 0, returns 1 for arg ≥>= 0
	sine of arg; arg expected in radians
	arg^2
	sqrt(arg)√arg

	tangent of arg; arg expected in radians
	$10^{**}arg$ 10^{arg}
	truncates arg to next lowest integer

Additional Difference Equations Topics

Input Sources in Difference Equations

Arithmetic Operators in Difference Equations

Non-Arithmetic Operators in Difference Equations

Constants in Difference Equations

Trig, Transcendental, and Calculus Functions in Difference Equations

Sampling Rate Conversion in Hypersignal

[@k downsampling, conversion,, sampling rate]

The fsconv.mac program is included with all Hypersignal-Macro series software packages.

When operating Hypersignal in user mode, the fsconv.mac program allows sampling rate conversion to appear as a virtual function when invoked from the Macro Manager menu. The fsconv.mac program prompts for an input time domain waveform and stretches or contracts the waveform to produce an output time domain with a different sampling rate. Input parameters include "multiply" and "divide" integer values, whose ratio, when multiplied against the input waveform sampling frequency, equals the output waveform sampling frequency. For example, to reduce an input waveform sampling rate by a factor of 10, a multiply integer of 1 should be entered, and a divide integer of 10. To increase sampling rate by a factor of 2, a multiply integer of 2 should be entered, and a divide integer of 1. See also comments in the fsconv.mac and dsplib.inc files.

The fsconv.mac program calls the f_fsconv.mac function, which can also be called from an outside application when Hypersignal is being operated in engine mode.

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Frequency Zoom in Hypersignal

[@k zoom,, frequency, baseband zoom,, frequency]

The fzoom.mac program is included only with Hypersignal-

Macro EX and Hypersignal-Acoustic software.

When operating Hypersignal in user mode, the fzoom.mac program allows frequency zoom to appear as a virtual function when invoked from the Macro Manager menu. The fzoom.mac program prompts for an input time domain waveform, and translates a portion of its overall bandwidth to the baseband, creating an output time domain waveform. Start (f_1) and stop (f_2) frequencies must be specified, which are entered as values normalized to 1/2 the sampling rate, or 0≤<=f_1≤<=1 and 0≤<=f_2≤<=1. The output time domain waveform filename must also be specified. See also comments inside the fzoom.mac and dsplib.inc files.

The fzoom.mac program calls the f_fzoom.mac function, which can also be called from an outside application when Hypersignal is being operated in engine mode.

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Minimum Phase in Hypersignal

[@k phase,, minimum]

The minphase.mac program is included only with Hypersignal-Macro EX and Hypersignal-Acoustic software

When operating Hypersignal in user mode, the minphase.mac program allows minimum phase to appear as a virtual function when invoked from the Macro Manager menu. The minphase.mac program prompts for an input time domain waveform, which should be a filter response or other impulse response, and creates a minimum phase version, also as a time domain waveform, with a user-specified output filename.

Minphase.mac attempts to minimize the overall group delay of the input impulse response, while leaving the frequency domain magnitude response unchanged. The minimum phase version should have the same frequency domain magnitude response (you can do an FFT Generation to check this), but minimized phase response, and an impulse response with energy concentrated near time=zero. For example, if you run minphase.mac on a Hypersignal FIR filter impulse response, you should see the resulting time domain impulse response looking somewhat like a "left-shifted" version of the original.

The minphase.mac program calls several macro language functions, including Difference Equations and FFT, during its course of processing. These other functions can also be called from an outside application when Hypersignal is being operated in engine mode.

When run on a waveform that already has minimum phase, such as an IIR filter impulse response, there should be essentially no change in the output from the input--in any of the magnitude, phase, or time domain impulse responses.

Examples of minimum phase usage include "what-if" scenarios to explore the effects of minimizing the delay in waveforms/sequences, reducing FIR filter lengths in real-time systems that are processing-intensive or have high sampling

rates, and creating minimum phase versions of impulse responses that have been realized through equalization (transfer function measurement of physical channels).

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Wavelet Transform in Hypersignal

[@k transform,, wavelet, continuous wavelet transform]

The wlt.mac program is included with both Hypersignal-Macro EX and Hypersignal-Acoustic for those who want to experiment with it. This program is still considered to be in an alpha-state, with improvements continuing to be added.

When operating Hypersignal in user mode, the wlt.mac program allows continuous wavelet transform to appear as a virtual function when invoked from the Macro Manager menu. The wlt.mac program prompts for input waveform filename, wavelet specification, output waveform filename, and number of scale values.

The wlt.mac program calls the f_wlt.mac function, which can also be called from an outside application when Hypersignal is being operated in engine mode.

In either mode, the wavelet specification can be one of the following:

A complex wavelet file (enter filename).

A real wavelet file (enter filename). In this case, a complex wavelet is created by the program, using a Hilbert Transform technique.

Offset and length of input filename (enter in samples, or points). In this case, the wavelet is taken from signal itself. For example, an entry of '76,32' in this field would cause the program to use a 32-point segment, starting at sample 76, from the input waveform. Note: when using this option, it is suggested that a somewhat periodic segment of the input waveform is chosen.

Some notes about the f_wlt.mac function:

A center frequency of the wavelet is automatically calculated using an FFT. Periodic wavelets lend themselves more to this calculation.

A continuous wavelet transform is performed using a chirp-z transform method developed at the Univ. of Illinois and Rice University. See the Hypersignal macro language wlt.mac file for more details and

algorithm/theory references and credits.

Output is produced compatible with Hypersignal frequency domain displays.

Currently, the function does not allow the input waveform to be segmented; as a result, large input waveforms are not yet practical.

One interesting observation to make using the wavsig.tim (input waveform) and gauss.tim (wavelet) files as examples in a comparison of the wavelet transform and a short time Fourier transform (STFT) computed using a baseband equivalent of the wavelet as a window function. This can be accomplished using the Gauss window option in the FFT Generation function (because the wavelet in this example is Gaussian in shape with a center frequency of $\pi/3$). For example, perform the following steps:

A description of the steps needed to make this comparison when operating Hypersignal in user mode is given below:

First, select the F3 key, which invokes the Macro Manager menu, and enter 'wlt' on the first line. This will cause a data-entry menu to appear prompting you for necessary entries. Use 'wavsig' as input waveform, 'gauss' as the wavelet filename, 'out' as the output filename, and number of scale values as '128'.

Second, select the FFT Generation menu item in Hypersignal and enter 'wavsig' as the input waveform. Enter the following parameters:

FFT order = 8 (FFT size = 256)

Framesize = 32 (samples)

Overlap = (Framesize–1)/Framesize∗100 (percent)

Window = Ga (Gaussian window)

Precision and Mag/Phase Approximation fields--if they should appear because you have no DSP hardware accelerator selected--as "don't cares" (they affect accuracy and speed somewhat, and can make a difference when exact results or large waveforms are required)

Third, enter 'wavsig' and 'out' in the 2-D Spectrograph display to see a comparison. You may want to enable the DUAL AUTOSCALE field in the User Setup area to see better comparison of amplitudes.

Overall, the choice of what type of analysis method to use (i.e. STFT, wavelet transform, chirp-z, etc.) is highly application- and data-dependent. Many experiments and iterations may be required to determine what works best. If indeed a wavelet transform method is to be used, further experimentation and research is needed to choose an effective wavelet.

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Schroeder Integration in Hypersignal

[@k integration,, Schroeder]

The schroed.mac program is included with Hypersignal-Acoustic software only.

When operating Hypersignal in user mode, the schroed.mac program allows Schroeder Integration to appear as a virtual function when invoked from the Macro Manager menu. The schroed.mac program prompts for an input and output time domain waveform filenames. Time domain

input data can be in either .TIM or .WAV formats. The function performs a reverse integration of the input waveform, creating a monotonically decreasing representation of the input waveform's "potential energy", or remaining energy, with respect to time. Display of Schroeder integration results should be made using the Waveform Display/Edit function, in either linear or log amplitude mode. By calculating a line-fit to the slope between time points in log mode of this display, reverberation-time measurements can be estimated. For example, the time required for energy to drop 60 dB along a reasonable line-fit could be used as an estimate for an RT60 measurement value.

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Equalization in Hypersignal

The equalize.mac program is included only with Hypersignal-Acoustic software.

When operating Hypersignal in user mode, the equalize.mac program allows equalization to appear as a virtual function when invoked from the Macro Manager menu. The equalize.mac program takes Spectrum Analyzer transfer function display (either analog signal or waveform file as input) and writes inverted linear magnitude output to a .FRQ file. This is done using the Spectrum Analyzer 'X' (inverted write) command, which is active only in Hypersignal-Acoustic software. The "zero dB" point used for inversion is the zero dB point on the transfer function display as default, or the current vertical position of the cursor, if displayed. The macro function then takes the inverted magnitude output, and the original phase file (which must be

created beforehand if the input was analog), and calculates real/imag components. It then calculates an inverse FFT to create a time domain impulse response which can be used as an FIR filter to equalize the original transfer function curve. As a sanity check, the macro function calculates the magnitude response of this impulse response, and displays it, along with the original, non-inverted magnitude response, in the Magnitude Display function, using the overlay mode for comparison and a display range of ±70 dB.

It is important to note that the length of the resulting FIR filter will be equal to the FFT size in the transfer function display, regardless of the size of the time domain (measurement) window in the transfer function display. Reasonable curve-matching may occur only with FFT sizes of 512 points or more, which means that it is unlikely that supported DSP hardware can implement filters of this size in real-time inside Hypersignal, since all Hypersignal instrumentation functions implement FIR filters using standard multiply-accumulate time domain techniques. For these reasons, you will probably need the Real-Time Convolver to apply the equalization filters in real-time, using supported DSP/analog I/O hardware.

When running Hypersignal-Acoustic under Windows, it is possible to set up and control two

separate DSP/analog I/O boards at once, using two copies of HS-Acoustic. The main system would control the measurement/instrumentation board, and the second copy the Real-Time Convolver board. In this way, the real-time Spectrum Analyzer can be used to monitor the effects of the equalization, and create new equalization filters, as needed.

After running equalize.mac, it may be desirable to run the resulting impulse response waveform (.tim file) through the minphase.mac program. The minimum phase function does not change the magnitude response, but it attempts to minimize the overall group delay. This may produce a more "normal-looking" impulse response, as well as shorten the number of taps required if the impulse response is to be downloaded and run as an FIR filter. However, phase preservation from the original, unequalized response will be destroyed.

The equalize.mac program has been extensively tested with frequency domain waveform files generated by the tefrd.exe and sysidrd.exe utilities, which convert real/imag output data files from Techron's Sound Lab software, and Ariel's SYSid software, to Hypersignal .FRQ and .PHZ formats. The tefrd.exe and sysidrd.exe utilities can be conveniently controlled using the tefrd.mac and sysidrd.mac macro

functions; these two utilities are included only with Hypersignal-Acoustic software.

When using a fixed-point DSP board at the same time as the equalize.mac function, it is suggested that the DSP ACCELERATOR field in the System Config menu be temporarily disabled. Even though the amount of error introduced by fixed-point FFT operations is small, it may be important in the equalization process, which is numerically sensitive.

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

MLS Stimulus & Response in Hypersignal

[@k stimulus & response,, MLS, maximum length sequence]

The mls.mac program is included only with Hypersignal-Acoustic software. This function is still considered to be in a beta state, with improvements continuing to be added.

When operating Hypersignal in user mode, the mls.mac program allows MLS Stimulus & Response to appear as a virtual function when invoked from the Macro Manager menu. The mls.mac program prompts for stimulus length, number of averages, output waveform filename, sampling frequency, a channel list, and stimulus amplitude and offset. It uses the currently installed hardware specified in the System Configuration menu ANALOG CONVERSION field, performs the stimulus & response operation, synchronously averages the results in the time domain, performs a cross-correlation operation to extract the impulse response, and then an FFT to calculate the frequency response. The macro program concludes by showing the frequency response in the

Magnitude Display, with vertical scale set to ±50 dB.

The mls.mac program calls the f_mls.mac, f_fft, and f_magdsp functions, which can also be called from an outside application when Hypersignal is being operated in engine mode.

Currently, f_mls.mac does not have an option to equalize out the response due solely to the measurement board being used. This option, and other "calibration mode" features, are expected to be added later.

MLS Stimulus & Response DSP Hardware Considerations

You must be aware of your DSP/analog I/O board memory size to make effective and reliable use of the mls.mac function. Some boards have a large amount of onboard DSP memory (either SRAM or DRAM) and let you run huge MLS stimulus sizes, some boards have very little memory, and most are in-between. Because Hypersignal-Acoustic supports so many types of hardware, the mls.mac function is very general in nature. See DSPower Reference Guide section xx, MLS Stimulus & Response, for more information.

Below is a short list of example boards, the amount of data memory available for use by Hypersignal when shipped with Signalogic-recommended memory amounts, and the maximum stimulus size. Memory amounts are in words

(natural wordlength of DSP device) and given in two values: Signalogic recommended amount that should be shipped with the board, and actual stimulus area available (excluding program area and input buffers). Memory amounts marked with an "*" can be packed double because the memory is accessible in words (16-bit values) as well as 32-bit values. Stimulus size values are in samples. See the Hypersignal-Macro Series Hardware Reference Guide for precise information about individual hardware types:

Table 6 MLS Stimulus Lengths vs. Hardware Type

Board	Total Mem	Stimulus Mem	Max Stimulus Size
Ariel DSP-32C	64k*	96k	32767
Ariel PC-56D+ ProPort	64k	20k	262143†
Ariel DSP-16	64k	20k	8191
CAC QUANTUMdsp	512k*	984k	262143
Innovative Integration PC31	128k	96k	32767
National Instruments AT-DSP2200	384k*	728k	262143
Signalogic Sig32C-8	128k*	192k	65535
Techron TEF20HI	48k	16k	262143†
TBS Tahiti	32k	12k	262143†

NOTE: †Indicates dynamic onchip generation of MLS output, which

avoids stimulus memory limitation. Currently active only for some DSP5600x hardware and some stimulus lengths.

For larger stimulus sizes, it is highly recommended that DSP hardware be obtained with at least an initial memory expansion option, and possibly the maximum amount of memory expansion allowable. In many cases, the DSP hardware manufacturer does not ship boards with much beyond 16k words of memory, for two reasons: in order to keep cost down, and because traditionally DSP programs have been known for compactness. Thus, it is important to know what is the memory amount status of your hardware.

Another important hardware consideration to be aware of when using mls.mac is memory speed. Memory that runs with wait-states can severely affect the maximum sampling rate possible when using the mls.mac function. For example, boards that provide DRAM for data buffers rather than SRAM can require as many as 4-6 wait-states to access the DRAM. DRAM is used to provide huge amounts of memory for reasonable cost, but there is always a tradeoff. Another example is the inexpensive multimedia class of hardware, which typically use slower SRAM (about 3 wait-states) in order to keep costs down. Generally, use of slow SRAM will cut maximum MLS sampling rates to about 25

kHz, and use of DRAM will cut maximum rates to about 16 kHz to 22 kHz. However, degradation depends also on the processor type and speed, which can mitigate decreases in maximum rates.

Additional Hypersignal Functions

FFT Generation in Hypersignal

Convolution/Correlation in Hypersignal

LPC Autocorrelation in Hypersignal

Recursive Filtering in Hypersignal

Power Spectra Generation in Hypersignal

Transfer Function Generation in Hypersignal

Inverse FFT Generation in Hypersignal

Difference Equations in Hypersignal

Sampling Rate Conversion in Hypersignal

Hypersignal-Macro EX Functions

Long FFT in Hypersignal

Long Convolution/Correlation in Hypersignal

Frequency Zoom in Hypersignal

Minimum Phase in Hypersignal

Wavelet Transform in Hypersignal

Hypersignal-Acoustic Functions

Schroeder Integration in Hypersignal

Equalization in Hypersignal

MLS Stimulus & Response in Hypersignal

Hypersignal Filter Design Overview

[@k filter design,, overview]

All Hypersignal-Macro series software packages contain reasonable digital filter design, offering from 75% to 85% of the capability of specialized filter design packages. Where Hypersignal-Macro series software performs better is in the actual use of filters--either in waveform file-based simulation, or in real-time. In addition to basic DSP simulation functions such as convolution and recursive filtering, a number of functions and displays are available that can utilize FIR and IIR filter coefficient files as input, including real-time instruments and functions such as sampling rate conversion. Filter coefficient files can be created in two ways:

Using Hypersignal-Macro series software built-in FIR Filter Design and IIR Filter Design functions, which offer both classical and arbitrary design methods.

Creating the filter coefficient files outside of Hypersignal, based on user-defined filter design methods or specifications.

Hypersignal Digital Filter Functions

Hypersignal Filter Design Overview
FIR Filter Design in Hypersignal
IIR Filter Design in Hypersignal
Filter DSP Code Generation in Hypersignal
Pole-Zero IIR Filter Display in Hypersignal
File-Based Filter Simulation in Hypersignal
"Snap-In" Real-Time Filtering in Hypersignal

FIR Filter Design in Hypersignal

[@k filter design,, Hypersignal]

All Hypersignal-Macro series software packages contain built-in Kaiser Window and Parks-McClellan design of lowpass, highpass, bandpass, and bandstop FIR filters. Differentiation and Hilbert Transform designs, which are a subset of Parks-McClellan design, may be specified. Also, arbitrary designs, such as multiband and irregular or log frequency passbands, can be specified and realized using the Parks-McClellan method. Selectable filter name, sampling frequency, center frequency, bandwidth, transition bandwidths, stopband attenuation, and passband ripple. Filter length is estimated by the software, but may be controlled explicitly. All designs are first displayed and can be measured in a graphical filter display prior to design; arbitrary designs can be further edited directly using the mouse. Resulting frequency responses can be compared in the graphical display to original

specifications and "tolerance regions".

Hypersignal filter design functions create as output a time domain impulse response waveform, and both an ASCII "filter coefficient file", which contains a numerator (pole) section and a denominator (zero) section (which is not used for FIR filters), and an ASCII "design goal file", which contains the filter specifications and the arbitrary frequency response, if any. Both of these latter two files can be created and manipulated directly, outside of Hypersignal, to achieve user-defined filters and numerically-specified design goals.

Current maximum FIR filter sizes are 4096 for Kaiser Window design, and 768 for Parks-McClellan design.

Hypersignal Digital Filter Functions

Hypersignal Filter Design Overview

FIR Filter Design in Hypersignal

IIR Filter Design in Hypersignal

Filter DSP Code Generation in Hypersignal

Pole-Zero IIR Filter Display in Hypersignal

File-Based Filter Simulation in Hypersignal

"Snap-In" Real-Time Filtering in Hypersignal

IIR Filter Design in Hypersignal

[@k filter design,, IIR]

All Hypersignal-Macro series software packages contain Butterworth, Chebyshev I and II, Bessel, and elliptic design of lowpass, highpass, bandpass, and bandstop IIR filters. Also, arbitrary designs, such as multiband and irregular or log frequency passbands, can be specified and realized using the Steiglitz method. Selectable filter name, sampling frequency, center frequency, bandwidth, transition bandwidths, stopband attenuation, and passband ripple or corner attenuation. In the case of Steiglitz design, the filter order and required minimum convergence error are also selectable. All designs are first displayed and can be measured in a graphical filter display prior to design; arbitrary designs can be further edited directly using the mouse. Resulting frequency responses can be compared in the graphical display to original specifications and "tolerance regions". Current maximum filter orders are 20 for Bessel designs, and 40 for Butterworth, Chebyshev I and II, elliptic and Steiglitz designs.

Hypersignal filter design functions create as output a time domain impulse response waveform, and both an ASCII "filter coefficient file", which contains a numerator (pole) section and a denominator (zero) section, and an ASCII "design goal file", which contains the filter specifications and the arbitrary frequency response, if any. Both of these files can be

created and manipulated directly, outside of Hypersignal, to achieve user-defined filters and numerically-specified design goals.

Current maximum IIR filter order is 40 for bandpass or bandstop, and 20 for lowpass or highpass.

Hypersignal Digital Filter Functions

Hypersignal Filter Design Overview

FIR Filter Design in Hypersignal

IIR Filter Design in Hypersignal

Filter DSP Code Generation in Hypersignal

Pole-Zero IIR Filter Display in Hypersignal

File-Based Filter Simulation in Hypersignal

"Snap-In" Real-Time Filtering in Hypersignal

Filter DSP Code Generation in Hypersignal

[@k code generation,, filters]

All Hypersignal-Macro series software packages contain assembly language and C source code generation for FIR and IIR filters. Both floating-point and fixed point DSP devices are supported, including the Analog Devices ADSP-21xx, AT&T DSP32C and DSP3210, the Motorola DSP5600x and DSP96002, the NEC 77220, and the Texas Instruments TMS320C25, and TMS320C3x and TMS320C4x. For fixed-point DSP chips, quantization of filter coefficients and delay storage

elements can be selected, ranging from 8 bits to the maximum word length of the DSP chip. A chip-independent quantization mode is also available, allowing quantization up to 32 bits. For IIR filters, cascade structure can be selected to be Direct Form, Modified Direct Form, and Balanced Pole-Zero. The quantization and cascade structure options are initialized by the software to optimum settings, depending on the filter type. Generated assembly language code is highly optimized and in loop form; quantization-related instructions are only inserted when necessary.

Also, simulated quantized impulse responses can be generated when working with fixed-point DSP devices. The quantized impulse responses can be processed with FFT Generation and compared to original filter design frequency responses in either the Magnitude Display function or in the filter design function, where tolerance region display and crosshair cursor measurement can be utilized to determine the effects of quantization.

Hypersignal Digital Filter Functions

Hypersignal Filter Design Overview

FIR Filter Design in Hypersignal

IIR Filter Design in Hypersignal

Filter DSP Code Generation in Hypersignal

Pole-Zero IIR Filter Display in Hypersignal

File-Based Filter Simulation in Hypersignal

"Snap-In" Real-Time Filtering in Hypersignal

Pole-Zero IIR Filter Display in Hypersignal

[@k filter display,, pole-zero]

All Hypersignal-Macro series software packages contain pole-zero display and analysis for IIR filters, created either from the built-in Hypersignal IIR filter design methods, or specified in user-defined IIR filter coefficient files. User-defined filters must be specified in Z-Domain 2nd-order factored form, or "biquad cascade" form, in IIR filter coefficient files. Z-Plane display is available for direct display of the filter coefficient file values. An inverse bilinear transformation is performed to allow normalized display of the filter transfer function in the S-Plane.

Hypersignal Digital Filter Functions

Hypersignal Filter Design Overview

FIR Filter Design in Hypersignal

IIR Filter Design in Hypersignal

Filter DSP Code Generation in Hypersignal

Pole-Zero IIR Filter Display in Hypersignal

File-Based Filter Simulation in Hypersignal

"Snap-In" Real-Time Filtering in Hypersignal

File-Based Filter Simulation in Hypersignal

[@k filtering, filter simulation]

For all Hypersignal-Macro series software packages, IIR filters designed in Hypersignal or created outside Hypersignal can be simulated using the Recursive Filtering function. The ASCII filter coefficient files offer both a convenient access point for you to take the results of Hypersignal filter design outside of Hypersignal, and for you to cause other Hypersignal filter-related functions to use your externally generated coefficients for further processing.

For all Hypersignal-Macro series software packages, FIR filters designed in Hypersignal or created outside Hypersignal can be simulated using the Convolution function.

Hypersignal Digital Filter Functions

Hypersignal Filter Design Overview

FIR Filter Design in Hypersignal

IIR Filter Design in Hypersignal

Filter DSP Code Generation in Hypersignal

Pole-Zero IIR Filter Display in Hypersignal

File-Based Filter Simulation in Hypersignal

"Snap-In" Real-Time Filtering in Hypersignal

"Snap-In" Real-Time Filtering in Hypersignal

[@k real-time filtering, filtering,, real-time]

For real-time versions of Hypersignal-Macro series software (all except Hypersignal-Macro NRT), IIR and FIR filters designed in Hypersignal or created outside Hypersignal can be "snapped-in" to the instrumentation and disk record/output functions for immediate real-time operation. In the case of the instrumentation displays, up to two filters can be enabled independently on dual-trace analog data, quantization can be selected, and cascade structure can be controlled from the ASCII filter coefficient files. Filtering takes place prior to all instrumentation calculations, such as transfer function and impulse response, allowing filters to become an intrinsic part of the measurement process, enabling selective frequency band analysis and/or delays. In the disk record/output and stimulus & response functions, one filter can be enabled on one or two channels; quantization can be selected and cascade structure can be controlled as with the instrumentation displays. In addition to the Code Generation function listed above, the routines used for real-time filtering, which include dynamic control over quantization, are available in all DSP Source Code Interfaces.

Hypersignal Digital Filter Functions

Hypersignal Filter Design Overview

FIR Filter Design in Hypersignal

IIR Filter Design in Hypersignal

Filter DSP Code Generation in Hypersignal

Pole-Zero IIR Filter Display in Hypersignal

File-Based Filter Simulation in Hypersignal

"Snap-In" Real-Time Filtering in Hypersignal

Hypersignal Interactive Display Overview

[@k display,, overview, interactive display,, overview]

When operated in user mode, Hypersignal-Macro series software packages offer several types of interactive displays which can be accessed directly.

When Hypersignal is opeating in engine mode, interactive displays can be accessed indirectly, in which case the user-interface is provided by the controlling application, and display and/or status results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

Interactive controls typically include zoom, translate, vertical scale, cursor readout, target window measurements, store to file, printout, frequency axis control, and more.

In the time domain Waveform Display/Edit display, rubberband and point-mode editing of .TIM and .WAV waveform files is active.

Also, in the case of real-time versions of Hypersignal-Macro series software, D/A playback is available from the waveform display/edit and joint time-frequency contour displays.

Hypersignal Display Functions

Hypersignal Interactive Display Overview

Time Domain Waveform Display/Edit in Hypersignal

Frequency Domain Magnitude and Phase Display in Hypersignal

Joint Time-Frequency Waterfall in Hypersignal

Joint Time-Frequency Contour in Hypersignal

Filter Design Display in Hypersignal

Pole-Zero Display in Hypersignal

Time Domain Waveform Display/Edit in Hypersignal

[@k display,, time domain, display,, .WAV, .WAV display]

All Hypersignal-Macro series software packages contain a time domain Waveform Display/Edit function which can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and display results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship

of Signalogic DSP Software Components, for more information).

The Waveform Display/Edit function is a general display that allows single-, dual-, or overlaid-trace display and editing of one or two waveforms. In the case of a single waveform, the data can be two-channel or complex. Also, waveforms other than time domain data can be displayed and edited. Both cut-and-paste and free-form "rubber band" editing modes are available. These modes can be freely switched between in editing sessions, and are fully enabled at all levels of zoom. Cut-and-paste operations include delete, copy, insert, and overwrite. Display controls include "current active trace" selection, framesize, zoom (and zoom-out), pan, amplitude scaling, random-access, target window and crosshair cursor measurement and readout, units, color and background, grid, and selection of vertical axis units, scaling factor, and maximum value.

In Hypersignal-Acoustic software, a log amplitude mode of display is also available, which depicts data as $10\log_{10}[x^2]$. All display and editing controls are active in this mode, which can easily be toggled on and off as needed.

Additionally, segments of displayed waveforms can be written to file or sent to the D/A output of the currently selected DSP/acquisition hardware. A built-in print function is also available

that allows screen dumps or publication-quality EMS memory regenerations to nearly 200 different printers.

Hypersignal Display Functions

Hypersignal Interactive Display Overview

Time Domain Waveform Display/Edit in Hypersignal

Frequency Domain Magnitude and Phase Display in Hypersignal

Joint Time-Frequency Waterfall in Hypersignal

Joint Time-Frequency Contour in Hypersignal

Filter Design Display in Hypersignal

Pole-Zero Display in Hypersignal

Frequency Domain Magnitude and Phase Display in Hypersignal

[@k display,, magnitude, display,, phase, magnitude display, phase display]

All Hypersignal-Macro series software packages contain frequency domain Magnitude and Phase Display functions which can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and display results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

The Magnitude Display function allows frame-by-frame display and inspection of frequency domain

magnitude data. The Phase Display function operates similarly, but for frequency domain phase data, and includes a group delay option and a "phase unwrapping" option, much like a Bode plot. These functions allow single-, dual-, or overlaid-trace display of one or two frequency domain waveforms.

For Magnitude Display, controls include log or linear amplitude or amplitude-squared display, "current active trace" selection, zoom (and zoom-out), pan, vertical amplitude limits, amplitude scaling, random-access, target window and crosshair cursor measurement and readout, units, color and background, grid, and selection of vertical axis units, scaling factor, and maximum value. For Phase Display, controls include wrapped phase, unwrapped phase, or group delay display format, "current active trace" selection, zoom (and zoom-out), pan, amplitude scaling, random-access, target window and crosshair cursor measurement and readout, units, color and background, grid, and selection of group delay smoothing factor.

For both types of display, the frequency axis can be specified to be linear or log. In the log mode, the log base, number of cycles to display, and minimum frequency axis value can be selected.

A built-in print function is also available that allows screen dumps or publication-quality EMS memory regenerations to nearly 200 different printers.

Hypersignal Display Functions

Hypersignal Interactive Display Overview

Time Domain Waveform Display/Edit in Hypersignal

Frequency Domain Magnitude and Phase Display in Hypersignal

Joint Time-Frequency Waterfall in Hypersignal

Joint Time-Frequency Contour in Hypersignal

Filter Design Display in Hypersignal

Pole-Zero Display in Hypersignal

Joint Time-Frequency Waterfall Display in Hypersignal

[@k frequency domain display, waterfall display]

All Hypersignal-Macro series software packages contain a frequency domain joint time-frequency waterfall display function which can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and display results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

The waterfall version of joint time-frequency display is sometimes referred to as a "landscape" or "spectrogram" display, and is labeled on the Hypersignal main menu as the 3-D Spectrogram. In a 3-D Spectrogram display, multiple

FFT output frames are displayed at once, with frequency along the horizontal axis, amplitude along the vertical, and time depicted as an "apparent", or z-axis going into the screen. Each successive FFT output frame is displayed behind the previous and hidden-line removal may be enabled, giving the overall appearance of a three-dimensional display. (Note: when changes between successive frames are relatively minor, the overall appearance of the display is similar to a cascading waterfall display.)

The 3-D Spectrogram allows single- or dual-trace display of one or two waveforms. In the case of a single waveform, the data can be two-channel. Controls include amount of data shown, log or linear amplitude or amplitude-squared display, vertical amplitude limits, zoom (and zoom-out), pan, random-access, target window measurement and readout, units, color and background, grid, angle of apparent incline (ranging from 5 to 90), and hidden-line removal and surface mesh options. Also, the frequency axis can be specified to be linear or log. In the log mode, the log base, number of cycles to display, and minimum frequency axis value can be elected.

A built-in print function is also available that allows screen dumps or publication-quality EMS memory regenerations to nearly 200 different printers.

Hypersignal Display Functions

Hypersignal Interactive Display Overview

Time Domain Waveform Display/Edit in Hypersignal

Frequency Domain Magnitude and Phase Display in Hypersignal

Joint Time-Frequency Waterfall in Hypersignal

Joint Time-Frequency Contour in Hypersignal

Filter Design Display in Hypersignal

Pole-Zero Display in Hypersignal

Joint Time-Frequency Contour Display in Hypersignal

[@k frequency domain display, contour display]

All Hypersignal-Macro series software packages contain a frequency domain joint time-frequency contour display function which can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and display results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

This display is sometimes referred to as a "spectrograph", and is labeled on the Hypersignal main menu as the 2-D Spectrograph. In a 2-D Spectrograph display, multiple FFT output frames are displayed at once, with frequency along the vertical axis, time along

the horizontal, and linear or log amplitude encoded either with different colors or gray shades. In this format, you are "looking down" at the frequency domain data--for this reason, the 2-D Spectrograph display is often referred to as a form of "contour plot." This type of display allows more joint time-frequency information to be displayed and to be viewable at one time than any other.

The 2-D Spectrograph allows single-, or dual-trace display of one or two waveforms. In the case of a single waveform, the data can be two-channel. Controls include "current active trace" selection, amount of data shown, vertical amplitude limits (which effectively control the amount of quantized amplitude in linear mode or the amount of quantized dB in log mode that is assigned to each color or gray shade), zoom (and zoom-out), pan, random-access, target window and crosshair cursor measurement and readout, units, color and background, and surface topology. Also, the frequency axis can be specified to be linear or log. In log mode, the log base, number of cycles to display, and minimum frequency axis value can be selected.

Additionally, segments of displayed waveforms can be sent to the D/A output of the currently selected DSP/acquisition hardware. A built-in print function is also available that allows screen dumps

or publication-quality EMS memory regenerations to nearly 200 different printers. Some color printers are supported. For black & white printers, gray level appearance is achieved through dot-density control.

Hypersignal Display Functions

Hypersignal Interactive Display Overview

Time Domain Waveform Display/Edit in Hypersignal

Frequency Domain Magnitude and Phase Display in Hypersignal

Joint Time-Frequency Waterfall in Hypersignal

Joint Time-Frequency Contour in Hypersignal

Filter Design Display in Hypersignal

Pole-Zero Display in Hypersignal

Filter Design Display in Hypersignal

[@k display,, filters]

All Hypersignal-Macro series software packages contain a graphical filter design and measurement display which can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and display results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

Proposed FIR and IIR filter designs are always displayed graphically in

Hypersignal prior to actual filter construction. Several measurement functions are available, which apply to both classical and arbitrary designs. Several editing functions are also available, which apply only to arbitrary designs. Measurement functions include target window and crosshair measurement and readout, grid, log/linear frequency and amplitude axes, overlay of actual and desired frequency responses, vertical scaling, and tolerance region display toggle. Editing functions include cut-and-paste and free-form rubberband modes, control over frequency point-density and "defined" and "undefined" regions, and initialization of response from menu specifications or design goal files. More information about filter design display is available under the Filter Design section, above.

Hypersignal Display Functions

Hypersignal Interactive Display Overview

Time Domain Waveform Display/Edit in Hypersignal

Frequency Domain Magnitude and Phase Display in Hypersignal

Joint Time-Frequency Waterfall in Hypersignal

Joint Time-Frequency Contour in Hypersignal

Filter Design Display in Hypersignal

Pole-Zero Display in Hypersignal

Pole-Zero Display in Hypersignal

[@k display,, pole-zero]

All Hypersignal-Macro series software packages contain a pole-zero analysis display which can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and display results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

The Pole-Zero Display function provides display and analysis for IIR filters, created either from the built-in Hypersignal IIR filter design methods, or specified in user-defined IIR filter coefficient files. User-defined filters must be specified in Z-Domain 2nd-order factored form, or "biquad cascade" form, in IIR filter coefficient files. Z-Plane display is available for direct display of the filter coefficient file values. An inverse bilinear transformation is performed to allow normalized display of the filter transfer function in the S-Plane.

Hypersignal Display Functions

Hypersignal Interactive Display Overview

Time Domain Waveform Display/Edit in Hypersignal

Frequency Domain Magnitude and Phase Display in Hypersignal

Joint Time-Frequency Waterfall in Hypersignal

Joint Time-Frequency Contour in Hypersignal

Filter Design Display in Hypersignal

Hypersignal Instrumentation & Measurement Overview

[@k overview,, instrumentation & measurement]

Real-time Hypersignal-Macro series software packages (all except Hypersignal-Macro NRT) contain several instrumentation and measurement functions which work in conjunction with supported DSP/analog hardware.

When Hypersignal is operating in user mode, instrument functions can be accessed directly. When Hypersignal is opeating in engine mode, instrument displays can be accessed indirectly, in which case the user-interface is provided by the controlling application, and display and/or status results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

Instrument functions typically download DSP executable files to the supported hardware, and then present interactive controls and/or displays that allow operation of the DSP code to be changed dynamically. Executable files for several supported DSP chip families are included with Hypersignal software; customization of these files or creation of new ones is possible. See section 1.2.3, DSP Source

Code Interface Software Summary, for more information.

Hypersignal Instrumentation & Measurement Functions

Hypersignal Instrumentation & Measurement Overview

Spectrum Analyzer in Hypersignal

Digital Oscilloscope in Hypersignal

Continuous Record/Output To/From Disk in Hypersignal

Stimulus & Response in Hypersignal

Signal Synthesis in Hypersignal

Spectrum Analyzer in Hypersignal

[@k instrumentation,, spectrum analyzer]

All real-time Hypersignal-Macro series software packages contain a version of a Spectrum Analyzer instrument function which can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and display results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

The Hypersignal Spectrum Analyzer can display single-, dual-, or overlaid-trace input from previously stored waveform files or any of the currently active analog channel inputs. Various combinations are possible,

including viewing the same analog channel in different domains, formats, or with different filtering options, and overlaying analog input with previously stored waveform file input. Selectable display data formats include time, linear/log magnitude, phase, transfer function, and impulse response (iFFT of transfer function). Averaging can be enabled. Operation from the DSP chip point of view is always in real-time up to the maximum sampling rate, but the rate of update of display frames to the video system of the host computer depends on the speed of the host I/O bus and video system. The maximum sampling rate possible depends on the DSP/acquisition board and DSP chip in use.

FIR and IIR filters created using Hypersignal filter design or supplied from outside of Hypersignal can be "snapped-in" to each trace independently and applied to analog input data in real-time (see "Snap-In" Real-Time Filtering, above). Maximum filter sizes and real-time operation with filters are dependent on the sampling rate and DSP/acquisition board in use.

Controls include timebase, trigger level, trace, and mode, vertical scaling and position, units, color and background, and crosshair cursor measurement, readout, and calibration (in engineering dB units). Trace displays can be captured to waveform file, or sent

to one of about 200 printers as a screen dump or publication-quality EMS memory regeneration. Fourier parameters such as FFT size, time domain framesize, and time domain window can be specified. Also, the software supports a "3-D map mode", which creates a waterfall type of display, in order to allow a time-history aspect in frequency data format display.

Frequency axes can be linear or log. In the log mode, the log base, number of cycles to display, and minimum frequency axis value can be selected. Also, 1/N octave band display can be enabled, which causes a "bar-graph" type of display.

For Hypersignal-Acoustic software only, trace displays can be inverted and stored to an FIR filter design goal file, and to magnitude and phase waveform files. An FIR filter can then be designed that approximates the inverted trace and applied in real-time to equalize the frequency response. Also for Hypersignal-Acoustic software only, real and imaginary transfer function data can be captured to waveform file, and a coherence display format is currently being installed.

When possible, the hardware resources of the DSP/acquisition board are under direct control. The sampling rate, analog domain anti-alias filtering (if programmable), input source, and hardware gain (if programmable) can be controlled

from the software. DSP/acquisition hardware supported by Hypersignal offer varied functionality, the best of which include maximum sampling rates of 10 MHz, up to 64 channels, programmable anti-alias filters, programmable input gain from 0 to 60 dB, and programmable input source selection ranging from line-level to microphone to AC power line.

Hypersignal Instrumentation & Measurement Functions

Hypersignal Instrumentation & Measurement Overview

Spectrum Analyzer in Hypersignal

Digital Oscilloscope in Hypersignal

Continuous Record/Output To/From Disk in Hypersignal

Stimulus & Response in Hypersignal

Signal Synthesis in Hypersignal

Digital Oscilloscope in Hypersignal

[@k instrumentation,, digital oscilloscope]

All real-time Hypersignal-Macro series software packages contain a version of a Digital Oscilloscope instrument function which can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and display results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software

Components, for more information).

The Hypersignal Digital Oscilloscope can display single-, dual-, or overlaid-trace time domain input from previously stored waveform files or any of the currently active analog channel inputs. Various combinations are possible, including viewing the same analog channel with different filtering options. Operation from the DSP chip point of view is always in real-time up to the maximum sampling rate, but the rate of update of display frames to the video system of the host computer depends on the speed of the host I/O bus and video system. The maximum sampling rate possible depends on the DSP/acquisition board and DSP chip in use.

FIR and IIR filters created using Hypersignal filter design or supplied from outside of Hypersignal can be "snapped-in" to each trace independently and applied to analog input data in real-time (see "Snap-In" Real-Time Filtering, above). Maximum filter sizes and real-time operation with filters are dependent on the sampling rate and DSP/acquisition board in use.

Controls include timebase, trigger level, trace, and mode, vertical scaling and position, units, and color and background. Trace displays can be captured to waveform file, or sent to one of about 200 printers as a screen dump

or publication-quality EMS memory regeneration.

When possible, the hardware resources of the DSP/acquisition board are under direct control. The sampling rate, analog domain anti-alias filtering (if programmable), input source, and hardware gain (if programmable) can be controlled from the software. DSP/acquisition hardware supported by Hypersignal offer varied functionality, the best of which include maximum sampling rates of 10 MHz, up to 64 channels, programmable anti-alias filters, programmable input gain from 0 to 60 dB, and programmable input source selection ranging from line-level to microphone to AC power line.

Hypersignal Instrumentation & Measurement Functions

Hypersignal Instrumentation & Measurement Overview

Spectrum Analyzer in Hypersignal

Digital Oscilloscope in Hypersignal

Continuous Record/Output To/From Disk in Hypersignal

Stimulus & Response in Hypersignal

Signal Synthesis in Hypersignal

Continuous Record/Output To/From Disk in Hypersignal

[@k record to file, record to disk, file record, disk record, data acquisition, acquisition,, waveform file, waveform files,, recording, files,, recording]

All real-time Hypersignal-Macro series software packages contain an Analog Conversion instrument function which can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and status and some data-monitoring results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

The Analog Convesion functions can perform continuous input and output of multichannel data to and from waveform files. Controls include sampling rate, hardware gain (if programmable), triggering (channel and level), software scaling and offsets, and amount of data. The maximum sampling rate possible depends on the number of channels in use, the selected DSP/acquisition hardware, and the speed of the host computer system (especially disk and I/O bus).

FIR and IIR filters created using Hypersignal filter design or supplied from outside of Hypersignal can be "snapped-in" and applied to analog input and output data in real-time (see "Snap-In" Real-Time Filtering, above). One real-time filter can be enabled for the input or output; for two-channel operation, the filter applies to both channels. Maximum filter sizes and real-time operation with

filters are dependent on the sampling rate and DSP/acquisition board in use.

When possible, the hardware resources of the DSP/acquisition board are under direct control. The sampling rate, analog domain anti-alias filtering (if programmable), input source, and hardware gain (if programmable) can be controlled from the software.

Hypersignal Instrumentation & Measurement Functions

Hypersignal Instrumentation & Measurement Overview

Spectrum Analyzer in Hypersignal

Digital Oscilloscope in Hypersignal

Continuous Record/Output To/From Disk in Hypersignal

Stimulus & Response in Hypersignal

Signal Synthesis in Hypersignal

Stimulus & Response in Hypersignal

[@k instrumentation, stimulus & response]

All real-time Hypersignal-Macro series software packages contain an Analog Conversion function capable of analog simultaneous input which can be used for stimulus & response measurement. The Analog Conversion function can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and status

and some data-monitoring results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

A single-channel or two-channel stimulus & response measurement function is available that allows simultaneous output of stimuli waveform and continuous recording of analog input to disk. Controls include sampling rate, hardware gain (if programmable), triggering (channel and level), software scaling and offsets, amount of data, and positive or negative stimulus delay. A positive delay directs recording to start after the onset of the stimulus; a negative delay directs recording to start before the stimulus. Operation from both the DSP chip and host computer point of view is always in real-time up to the maximum sampling rate. The maximum sampling rate possible depends on the number of channels in use, the selected DSP/acquisition hardware, and the speed of the host computer system (especially the disk and I/O bus).

FIR and IIR filters created using Hypersignal filter design or supplied from outside of Hypersignal can be "snapped-in" and applied to analog input data in real-time (see "Snap-In" Real-Time Filtering, above). One real-time filter can be enabled for the input only; for two-channel operation, the filter applies to both channels.

Maximum filter sizes and real-time operation with filters are dependent on the sampling rate and DSP/acquisition board in use.

In Hypersignal-Acoustic software, a single-step MLS Stimulus & Response macro program (mls.mac) is available that generates a pseudo-random maximum length sequence (MLS) stimulus, outputs the stimulus while continuously recording the response of the system under test, and then performs cross-correlation and FFT operations to calculate the impulse and frequency responses of the system. The mls.mac program offers averaging, amplitude and offset control, and other features.

See Also

MLS Stimulus & Response in Hypersignal

Hypersignal Instrumentation & Measurement Functions

Hypersignal Instrumentation & Measurement Overview

Spectrum Analyzer in Hypersignal

Digital Oscilloscope in Hypersignal

Continuous Record/Output To/From Disk in Hypersignal

Stimulus & Response in Hypersignal

Signal Synthesis in Hypersignal

Signal Synthesis in Hypersignal

[@k continuous signal generation, synthesis,, signal]

All real-time Hypersignal-Macro series software packages contain an

Analog Conversion function capable of continuous, repetitive analog output which can be used for signal generation. The Analog Conversion function can be accessed directly when operating in user mode, and indirectly when Hypersignal is in engine mode. In the latter case, the user-interface is provided by the controlling application, and status and some data-monitoring results are sent to the engine via the Pipe.386 VxD (see section 1.1, Relationship of Signalogic DSP Software Components, for more information).

Continuous signal synthesis in Hypersignal is implemented by output looping in the Analog Conversion function. By creating short signal/function waveforms with the Difference Equations function, and then using these waveforms as input to the Analog Conversion function, a continuous signal generator can be achieved when the ACQ/GEN field is set to 'GL'. Some guidelines to follow when specifying a Difference Equation:

- waveform lengths (as specified in the Difference Equations # OF SAMPLES field) should be an integer multiple of the function period; if you are in doubt about the function period, use the Waveform Display/Edit option to measure it exactly
- the overall waveform length should be anywhere from a few hundred samples to less than 6000 samples; only one or just a few periods of the function waveform is not recommended, because of the overhead required by the DSP software when "wrapping around" to repeat the function. And, definitely, long waveforms are not recommended because they become

split into buffer segments due to DSP board memory limitations, defeating the continuous mechanism altogether

- the length of the waveform, in samples (and multiplied by the number of channels), should be divisible by two
- watch out for sampling frequency when designing the difference equation; most hardware only allows certain sampling frequencies, and if you base your calculations on a sampling frequency the hardware cannot achieve, then there is likely to be a discontinuity when the output waveform repeats

Hypersignal Instrumentation & Measurement Functions

Hypersignal Instrumentation & Measurement Overview

Spectrum Analyzer in Hypersignal

Digital Oscilloscope in Hypersignal

Continuous Record/Output To/From Disk in Hypersignal

Stimulus & Response in Hypersignal

Signal Synthesis in Hypersignal

Macro Language: Hypersignal Programmability

[@j Hypersignal macro language, @k programmability,, macro language]

The Hypersignal macro language can be described as a "watered-down combination of Pascal and C". As with other languages, variables and programming constructs are available to store results, isolate often-used segments of code, invoke standard mathematical, file, and I/O functions, repeat segments of code, and make decisions based on results--in general, to solve

problems and allow "what-if" experimentation. Unlike other languages, the macro language provides a direct "hot-line" to Hypersignal functions and displays, not only allowing them to be invoked and combined in any sequence, but to accept macro language variables as input parameters. This allows the numerous functions and displays in Hypersignal to appear as a library of callable routines. Some overall features of the Hypersignal macro language:

- Wide range of numerical and string operations, including 32-bit floating-point, 32-bit integer, and 40-character string operations.
- If-then, for-loop, and nested procedure constructs.
- Built-in file I/O which "knows" about .TIM and .WAV waveform file formats.
- "Interactive commands" which can be issued inside displays and instruments.
- Temporary or permanent transition to manual control from executing macro language programs.
- Built-in Hypersignal macro language

compiler allows macro language programs to be used and distributed in exectuable form; a "make mode" is available to avoid compilation of source code files that have not changed.

Following is information about important macro language features, includng partial lists of operators, commands, and functions, and also a source code listing example:

Related Macro Language Topics

Macro Language: Hypersignal Programmability

Macro Record Mode

Hypersignal Macro Programs

Creating Macro Language Programs and Functions

Macro Language Variables, Constructs, and Procedures

Macro Language Operators

Macro Language Built-In Functions

Macro Language Interactive Commands

Macro Language Source Code Example

Macro Record Mode

[@k recording,, macros]

For repetitive tasks, a "record mode" can be enabled, in which the software records manual function and display steps to a macro source file, which can be "played back" afterwards.

Related Macro Language Topics

Macro Language: Hypersignal Programmability

Macro Record Mode

Hypersignal Macro Programs

Creating Macro Language Programs and Functions

Macro Language Variables, Constructs, and Procedures

Macro Language Operators

Macro Language Built-In Functions

Macro Language Interactive Commands

Macro Language Source Code Example

Hypersignal Macro Programs

[@k macro language programs]

Several ready-to-execute macro programs are provided with Hypersignal software. To achieve manually in Hypersignal, these would require multiple steps, as well as error-prone and distracting "thinking": intermediate calculation and what function sequence to use. The standard Hypersignal-Macro package includes sampling rate conversion and several examples of operations such as cut-and-paste waveform editing, D/A output from displays, etc. The Hypersignal-Macro EX package includes more powerful macro programs, such as frequency zoom and single-step stimulus & response. Hypersignal-Acoustic software includes still more macro programs, such as Schroeder integration and utilities to read

Ariel SYSid and Techron SoundLab output files.

See also Programs, Template Files, and Function Files in section 1.1, Relationship of Signalogic DSP Software Components.

Related Macro Language Topics

Macro Language: Hypersignal Programmability

Macro Record Mode

Hypersignal Macro Programs

Creating Macro Language Programs and Functions

Macro Language Variables, Constructs, and Procedures

Macro Language Operators

Macro Language Built-In Functions

Macro Language Interactive Commands

Macro Language Source Code Example

Creating Macro Programs and Functions

[@k function files,, macro language, programs,, macro language]

A macro language program can consist of one or more macro language source files. A common approach is to create a main macro language *program* source file, with a descriptive name to be used as the program name for user-interface purposes, and one or more macro language *function* source files which contain procedures. The function files are then included at compile-time using the macro language include-file construct. Many of the ready-to-execute macro language programs that

come with Hypersignal-Macro EX and Hypersignal-Acoustic take this approach. DSPower series software, when using a Hypersignal-Macro series software package in engine mode, use this technique. See DSP Engines and Programs, Template Files, and Function Files in section 1.1, Relationship of Signalogic DSP Software Components.

Macro language source files can be created using traditional programming techniques (a text editor!), the built-in Hypersignal Macro Record Mode, or a combination of both. Source files are then compiled by the built-in Hypersignal-Macro compiler, which in turn creates a single "macro object file" representing the macro program. This object file is what Hypersignal actually executes when the program is run, when operating in either user mode or engine mode.

After compiling once, only the macro object file need be accessed to execute a macro program repeatedly. A "compiler mode" can be set in the Macro Manager menu that controls the circumstances under which compiling occurs. A "make" setting automatically compiles source files only if they have been changed.

Related Macro Language Topics

Macro Language: Hypersignal Programmability

Macro Record Mode

Hypersignal Macro Programs
Creating Macro Language Programs and Functions
Macro Language Variables, Constructs, and Procedures
Macro Language Operators
Macro Language Built-In Functions
Macro Language Interactive Commands
Macro Language Source Code Example

Macro Language Variables, Constructs, and Procedures

[@k variables,, macro language, procedures,, macro language]

As with most high-level languages, variables must be "declared" to be one of several types before use, procedures can be defined before the "main body" of source code, mathematical operators and functions can be invoked, and various conditional and looping constructs are available. However, the macro language approach also allows unique ability, such as a simplified means of invoking native Hypersignal-Macro series functions and displays, and graceful transitions between "interactive" and "automatic" modes. For example, the system can stop in the middle of an automated test procedure, wait for the operator to make measurements and perform operations, and then resume the test procedure.

Variables and constants can be 32-bit integer, 32-bit floating-point (real), or 40-character string. There is also a boolean pseudo-type

which is essentially treated as an integer. Constants can be indicated in decimal, floating-point, or hexadecimal.

Related Macro Language Topics

Macro Language: Hypersignal Programmability

Macro Record Mode

Hypersignal Macro Programs

Creating Macro Language Programs and Functions

Macro Language Variables, Constructs, and Procedures

Macro Language Operators

Macro Language Built-In Functions

Macro Language Interactive Commands

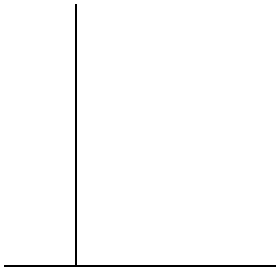
Macro Language Source Code Example

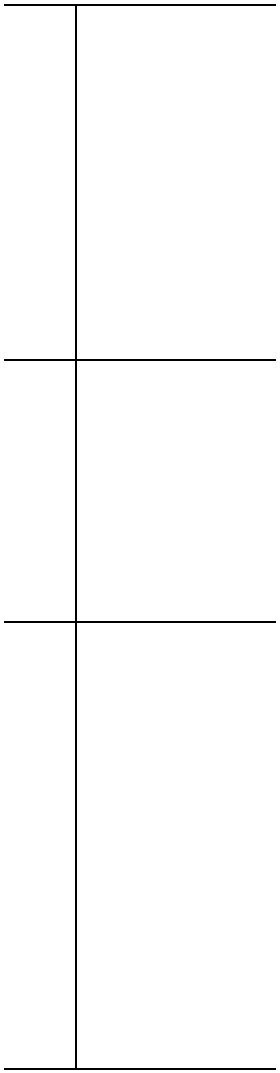
Macro Language Operators

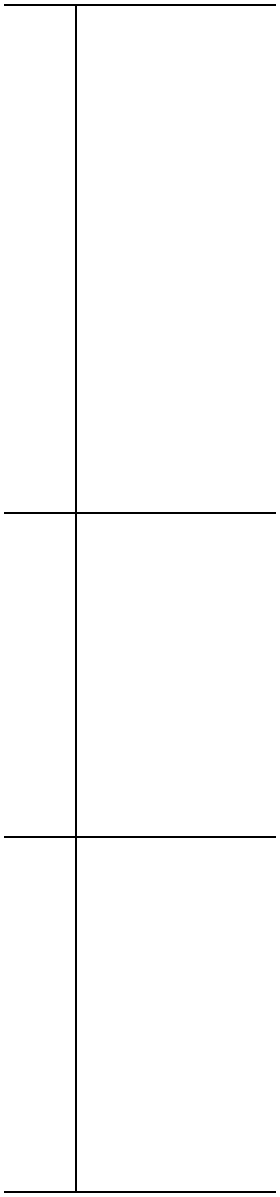
[@k operators,, macro language]

Below is a partial list of Hypersignal macro language operators:

Table 7 Partial List of Hypersignal Macro Language Operators







Related Macro Language Topics

Macro Language: Hypersignal Programmability

Macro Record Mode

Hypersignal Macro Programs

Creating Macro Language Programs and Functions

Macro Language Variables, Constructs, and Procedures

Macro Language Operators

Macro Language Built-In Functions

Macro Language Interactive Commands

Macro Language Source Code Example

Macro Language Built-In Functions

[@k functions,, macro language]

Several built-in functions are available in the Hypersignal macro language. Moreover, the language contains "interactive commands" that allow waveform and instrument displays to be controlled from macros. Below is a partial list of macro language functions and interactive commands. "Ident"

refers to an identifier: a constant or variable (real, integer, or string):

Table 8 Partial List of Macro Language Built-In Functions

Abort

Abs(var)	

Delay(N)	
D	

isplay(ident)	

DOS Shell	

Exit	

Exp(var)	

var = GetDataValue (filename, N)	

var = GetDateTime()	

var = GetHeaderValue (filename, N)	

var = GetUserSetup(N)	

Ln(var)	
Log(var)	

PutDataValue (filename, N, value)	

PutHeaderValue (filename, N, val	

u **e** **)**	

Round(var)	

Sqr(var)	
Sqrt(var)	

TenX(var)	
Trace(N)	

Trunc	

(var)	
WaitFor	

DateTime(DT)	

WaitForKey(K)

Related Macro Language Topics

Macro Language: Hypersignal Programmability

Macro Record Mode

Hypersignal Macro Programs

Creating Macro Language Programs and Functions

Macro Language Variables, Constructs, and Procedures

Macro Language Operators

Macro Language Built-In Functions

Macro Language Interactive Commands

Macro Language Source Code Example

Macro Language Interactive Commands

[@k interactive commands,, macro language]

Several "interactive commands" are available in the Hypersignal macro language. These commands allow waveform and instrument displays to be controlled from macros. In the partial list of interactive commands below, "Ident" refers to an identifier: either a constant or variable (real, integer, or string):

Table 9 Partial List of Macro Language Interactive Commands

	Invokes cut-and-

		paste delete prompt.
		Macro execution is delayed for N msec.
		Makes the target window smaller, or moves the crosshair cursor down.
		Invokes cut-and-paste

		insert prompt.
		Moves the target window or the crosshair cursor to the left.
		Stops macro execution and returns controls to manual mode. Execution of the macro

		program is halted until the current interactive display is exited (manually) by the user. All controls are active in manual mode.
		Pans backward in waveform displays.
		Pans

		forward in waveform displays.
		Reduces amplitude scaling in waveform displays.
		Increases amplitude scaling in waveform displays.
		Invokes the pop-up print menu in waveform

		displays
		Moves the target window or the crosshair cursor to the right.
		Enables or disables interactive command trace display.
		Returns to previous level of zoom.

	Makes the target window larger, or moves the crosshair cursor up
	Stops macro execution until the system date and time reach the value of DT.
	Stops macro exec

		utio n until the key 'K' is pres sed at the keyb oard.
		In wav efor m displ ays, exec utes ZO OM inter activ e displ ay com man d.

Related Macro Language Topics

Macro Language: Hypersignal Programmability

Macro Record Mode

Hypersignal Macro Programs

Creating Macro Language Programs and Functions

Macro Language Variables, Constructs, and Procedures

Macro Language Operators

Macro Language Built-In Functions

Macro Language Interactive Commands

Macro Language Source Code Example

Macro Language Source Code Example

[@k example,, macro language source, source code example,, macro language]

Below is a partial sample macro file source listing taken from the sampling rate conversion macro included with Hypersignal-Macro software. The listing has been simplified for brevity:

```
; parameters passed to this macro from the "Macro Manager" menu are:
; #1: input waveform name
; #2: M; integer amount to multiply input sampling rate by
; #3: N; integer amount to divide input sampling rate by
; #4: output waveform name

; declare and calculate a few variables that will be needed
Integer InSampFreq InLen InFrameSize InCutOffFreq CutOffFreq D
Integer StretchSampFreq StretchLen StretchFrameSize
Integer OutSampFreq OutLen OutFrameSize OutCutOffFreq

InSampFreq = GetHeaderValue(#1, Fs)  ; get values from input header
InLen = GetHeaderValue(#1, Len)
InFramesize = GetHeaderValue(#1, Fr)
InCutOffFreq = InSampFreq div 2      ; "Nyquist" rate for input waveform

StretchSampFreq = InSampFreq * #2    ; calculate "stretched" values
StretchLen = InLen * #2
StretchFramesize = InFramesize * #2

OutSampFreq = StretchSampFreq div #3 ; calculate final converted values
OutLen = StretchLen div #3
OutFramesize = StretchFramesize div #3
OutCutOffFreq = OutSampFreq div 2    ; "Nyquist" rate for output

Difference Equations                 ; apply difference equation to stretch
                                     input
    stretch                          ; output waveform name
    y[n]=x1[n\#2]                    ; #2 is second input parameter
    blank
```

```
    blank
    blank
    #1                              ; input waveform
    none
    @StretchLen                     ; length of input waveform * parameter
                                    #2
    @StretchSampFreq                ; sampling freq of input * parameter #2
    @StretchFramesize               ; framesize of input * parameter #2
endfunc

if InCutOffFreq<OutCutOffFreq then  ; design FIR filter to interpolate/anti-
                                    alias
    CutOffFreq = InCutOffFreq       ; between sample points stretched
else                                ; waveform--use whichever cutoff
    CutOffFreq = OutCutOffFreq      ; frequency is less
endif

FIR Filter Construction             ; design interpolation anti-aliasing filter
    interp                          ; filter name
    kw                              ; design method (KW = Kaiser Window)
    @StretchSampFreq                ; sampling frequency (in Hz)
    0                               ; center frequency (in Hz)
    @CutOffFreq                     ; bandwidth (in Hz)
    0                               ; left transition bandwidth (in Hz)
    250                             ; right transition bandwidth (in Hz)
    50                              ; stopband attenuation (in dB)
    auto                            ; let software choose number of taps
    1                               ; passband ripple (in dB)
    $RUN                            ; leave menu editor, execute function
    $C                              ; construct filter
endfunc

Convolution/Correlation             ; apply interpolation/anti-aliasing filter
    stretch                         ; input waveform name
    interp                          ; filter name
    stretchf                        ; output waveform name
    –                               ; '–' invokes convolution
endfunc

D = GetHeaderValue(interp, Len)     ; get interp filter length

Difference Equations                ; use difference equation to decimate
                                    ; output and remove filter delay
    #4                              ; input parameter #4 is output filename
    y[n]=x1[@D+n*#3]                ; input parameter #3 is divide factor N
    ""                              no entry
    ""
    ""
    stretchf                        ; input waveform "x1"
    none                            ; x2 not used
```

```
    @OutLen                 ; final length of output waveform
    @OutSampFreq            ; final sampling freq of output waveform
    @OutFramesize           ; final framesize of output waveform
endfunc
```

Related Macro Language Topics

Macro Language: Hypersignal Programmability

Macro Record Mode

Hypersignal Macro Programs

Creating Macro Language Programs and Functions

Macro Language Variables, Constructs, and Procedures

Macro Language Operators

Macro Language Built-In Functions

Macro Language Interactive Commands

Macro Language Source Code Example

File Import/Export Functions in Hypersignal

[@k importing waveform files, exporting waveform files]

All Hypersignal-Macro series software packages contain a powerful, mature "File Acquisition" function, which has evolved over the years in response to the demands of many different types of customers. This function can acquire binary and ASCII data in decimal, hexadecimal, floating-point, and various DSP manufacturer formats. Data sources include disk files, RS-232 serial channel, and direct PC memory access. Data can be unsigned, μ-Law or A-Law, complex, and multichannel; control or other non-data information contained in individual samples can

be stripped. Sample size ranges from 8 bits to 32 bits in 4 bit, an "interleave factor" can be specified when dealing with multichannel data, the mean can be subtracted from incoming data, and initial data (such as a header from another program) can be discarded from the beginning of the input data. Hypersignal waveform files can also be exported to ASCII decimal, floating-point, and hexadecimal formats.

The File Acquisition function can be invoked from the Utilities menu when Hypersignal is operating in user mode, and from the f_fileac macro language function when Hypersignal is being operated in engine mode.

Import/Conversion Topics

File Import/Export Functions in Hypersignal

Tim2Wav and Wav2Tim Waveform File Conversion Functions in Hypersignal

Tim2Wav and Wav2Tim Waveform File Conversion Functions in Hypersignal

[@k conversion,, .WAV files, conversion,, .TIM files]

The tim2wav.mac and wav2tim.mac programs are included with all Hypersignal-Macro series software packages, and can be invoked from the Macro Manager menu when Hypersignal is operating in user mode. The tim2wav.mac and wav2tim.mac

programs call the f_tm2wv and f_wv2tm functions, respectively. Use the 'F3' key to enter the Macro Manager menu, and then enter the program name. Both programs prompt for input and output waveform filenames, with no extension.

The f_tm2wv and f_wv2tm functions can also be called from DSPower-Block Diagram when Hypersignal is operating in engine mode.

The f_tm2wv and f_wv2tm functions convert waveform files between Hypersignal .TIM format and Microsoft .WAV format.

Regardless of operating mode, all Hypersignal time domain functions, including Difference Equations, Convolution, FFT, D/A playback, etc. can read .WAV files, either 8-bit or 16-bit, directly as input. For these functions, a '.WAV' extension must be entered or supplied explicitly to the appropriate input waveform name fields. *The only exception* is waveform editing with 8-bit .WAV files. Only 16-bit .WAV files are allowed in Hypersignal waveform editing.

When operating Hypersignal in user mode, if you need to reconvert a .WAV file after processing by Hypersignal, or convert any other Hypersignal waveform file to .WAV format, use the Hypersignal tim2wav.mac program, as explained above.

NOTE: At no time do Hypersignal functions generate .WAV files automatically as output, including the PutHeaderValue and PutDataValue macro language commands. This is because the .WAV format file header is limited, and cannot support all the parameters output in typical Hypersignal waveform files. It is possible to add application-specific information to .WAV files, but this capability is not encouraged in the Microsoft documentation, nor is it found in widespread use. For more information about these limitations, and the .WAV file format in general, see Signalogic DSP Software System Guide section 4.2, Microsoft .WAV Waveform File Format, and chapter 8 in the Microsoft Windows Multimedia Programmer's Reference.

File Import/Conversion Topics

File Import/Export Functions in Hypersignal

Tim2Wav and Wav2Tim Waveform File Conversion Functions in Hypersignal

External Program Control In Hypersignal

[@k controlling external programs, external programs]

External programs can be executed from a Hypersignal-Macro series program in three ways:

From one of six (6) different user-defined menu labels when

Hypersignal is operating in user mode. These menu labels can be customized and associated with various external programs using the Hypersignal the Utilities menu selection Ext Program Config.

From macro language programs which have been started from the Macro Manager menu when Hypersignal is operating in user mode. This technique is used by numerous macro programs included with Hypersignal-Macro EX and Hypersignal-Acoustic software, which need to execute protected-mode DOS programs in several cases.

From macro language programs which have been started by another application when Hypersignal is operating in engine mode. This technique is used by DSPower-Block Diagram when using a Hypersignal package as a DSP Engine

Real-mode or protected-mode DOS programs can be executed in all cases, and Windows programs can be executed if Hypersignal is being operated in engine mode by a Windows application. In the latter two techniques listed above, up to 10 command-line parameters can be passed from the macro language program to the external program.

Related Hypersignal System Consideration Topics

External Program Control in Hypersignal

User-Interface in Hypersignal

Online Help in Hypersignal

Graphics and Mouse Support in Hypersignal

80x87 Math Coprocessor Support in Hypersignal

Printer Support in Hypersignal

Documentation Supplied with Hypersignal

User-Interface in Hypersignal

[@k Hypersignal user-interface]

The Hypersignal-Macro series user-interface contains many aides and utilities to facilitate use of the program. Moreover, the maturity of the software reflects the countless suggestions and criticisms made by customers over the past few years. All functions and displays are part of the same integrated environment, and produce and accept compatible waveform data. The basic interface is menu-driven, with individual

functions and displays being controlled by "set-up parameters". This approach has enabled Hypersignal to gain popularity and success as a "lab-bench" tool known for its ease-of-use and environment conducive to freedom-of-thought.

In engine mode, all visual aspects of the Hypersignal user-interface are deactivated. In engine mode, all menus, data-entry fields, etc. receive simulated keyboard input from the macro language "external application pipe" facility (see Pipe.386 VxD in section 1.1, Relationship of Signalogic DSP Software Components).

Related Hypersignal System Consideration Topics

External Program Control in Hypersignal

User-Interface in Hypersignal

Online Help in Hypersignal

Graphics and Mouse Support in Hypersignal

80x87 Math Coprocessor Support in Hypersignal

Printer Support in Hypersignal

Documentation Supplied with Hypersignal

Online Help in Hypersignal

[@k help,, online]

When operating a Hypersignal-Macro series software package in user mode, a native online Help facility provides over 300 pages of online Help, selectable from menus

and displays in the software. The native Hypersignal online Help facility is context-sensitive, refers to appropriate sections in the documentation, and can be easily controlled using the keyboard or mouse.

Related Hypersignal System Consideration Topics

External Program Control in Hypersignal

User-Interface in Hypersignal

Online Help in Hypersignal

Graphics and Mouse Support in Hypersignal

80x87 Math Coprocessor Support in Hypersignal

Printer Support in Hypersignal

Documentation Supplied with Hypersignal

Graphics and Mouse Support in Hypersignal

When operating a Hypersignal-Macro series software package in user mode, over 100 different video hardware types, including CGA, EGA, VGA, Hercules, various 800x600 and 1024x768 video boards, and several specialized hardware types and resolutions, are supported with the MetaWindow graphics drivers. Support is also available for NEC PC98xx-series machines used in Japan. The MetaWindow drivers also include support for several types of mice, either through the mouse manufacturer's memory-resident driver or directly through the serial port. All Hypersignal menus and

displays can be easily controlled using either the mouse or keyboard. A "hardware probe" is conducted by the software to provide best-guesses for graphics and mouse driver selection.

When a Hypersignal-Macro series software packge is being operated in DSP Engine mode, the MetaWindow drivers are not loaded.

Related Hypersignal System Consideration Topics

External Program Control in Hypersignal

User-Interface in Hypersignal

Online Help in Hypersignal

Graphics and Mouse Support in Hypersignal

80x87 Math Coprocessor Support in Hypersignal

Printer Support in Hypersignal

Documentation Supplied with Hypersignal

80x87 Math Coprocessor Support in Hypersignal

Hypersignal-Macro series software packages contain automatic detection and utilization for 80x87 math processors. The automatic detection process can be manually overridden if necessary (see section 3.2.5, Installing and Configuring Hypersignal Software for User Mode Operation. A math processor in your system is strongly recommended: Hypersignal-Macro series software packages use 8-byte and 10-byte precision in several

instances, and the presence of a math processor will increase the speed of the software in general and several design and simulation functions, such as filter design and difference equations, in particular. Speed increases can be substantial (from 4 to 6 times). If no math processor is present in your system, Hypersignal-Macro series software packages will exactly emulate all 80x87 calculations using software routines, producing no difference in operating results (but executing much slower).

Related Hypersignal System Consideration Topics

External Program Control in Hypersignal

User-Interface in Hypersignal

Online Help in Hypersignal

Graphics and Mouse Support in Hypersignal

80x87 Math Coprocessor Support in Hypersignal

Printer Support in Hypersignal

Documentation Supplied with Hypersignal

Printer Support in Hypersignal

Hypersignal-Macro series software packages contain built-in support for about 200 dot-addressable printers, ranging from various Epson and compatibles to LaserJet and compatibles. When being operated in user mode, waveform and instrumentation displays in Hypersignal software can be printed as screen dumps or as

publication-quality EMS memory regenerations. Controls include print image dimensions, orientation, background, printer resolution (dpi), and number of copies.

Related Hypersignal System Consideration Topics

External Program Control in Hypersignal

User-Interface in Hypersignal

Online Help in Hypersignal

Graphics and Mouse Support in Hypersignal

80x87 Math Coprocessor Support in Hypersignal

Printer Support in Hypersignal

Documentation Supplied with Hypersignal

Documentation Supplied with Hypersignal

Hypersignal-Macro series software documentation is composed of Users Manual, Hardware Reference Guide, Software Reference Guide, and Tutorial, totaling more than 1200 pages. The Tutorial is illustrated with numerous menu and waveform display screen examples. The Hardware Reference Guide discusses all DSP/acquisition hardware and systems supported by the software individually and in detail, and also provides standardized analog and performance specification tables for each hardware type.

Related Hypersignal System Consideration Topics

External Program Control in Hypersignal

User-Interface in Hypersignal

Online Help in Hypersignal

Graphics and Mouse Support in Hypersignal

80x87 Math Coprocessor Support in Hypersignal

Printer Support in Hypersignal

Documentation Supplied with Hypersignal

1.2.2 DSPower-HWLib Software Summary

DSPower-HWLib software consists of two (3) DLLs and various C/C++ source code examples, include files, online Help files, and support programs. The DSPower-HWLib DLL itself is a Windows DLL that offers a C/C++ interface to supported hardware. Low-level functions include board and individual processor control, program file download, memory transfer, and semaphore management. High-level functions include acquisition to waveform file, D/A playback from file or continuous function generation, stimulus & response, various DSP functions, and more.

DSPower-HWLib software also includes the Engine Manager and Hardware Manager DLLs (enmgr.dll and hwmgr.dll), which are used to control DSP Engines,

and maintain current configuration for supported DSP/analog hardware types, respectively. See Signalogic DSP Software Users Guide chapters 7 and 8, Using Engine Manager and Using Hardware Manager.

The Engine Manager allows multiple copies of DSP Engines to be opened and controlled. Arbitrary engine functions can be executed using the DSExecEngineFunction call.

Hardware Manager maintains a Hardware Setup File which includes such parameters as I/O and memory base addresses, analog I/O voltage levels, memory amounts and architecture, processor clock rates, and more. It also offers the DSShowHardwareSelectorDlg call, which displays a Hardware Selector dialog box allowing user-specified selection of hardware type and configuration parameters.

1.2.3 DSP Source Code Interface Software Summary

The DSP Source Code Interfaces provide access to DSP source for nine (9) different DSP devices, including math and DSP routines, analog I/O drivers, and real-time processing structures used by the Hypersignal packages. They can be used to customize Hypersignal operation, or operation of some DSPower-HWLib software high-

level operations when a Hypersignal package is used as the DSP Engine.

Modification of, or addition to, the DSP Source Code Interface for a specific DSP chip allows customization of the standard executable DSP files delivered with Hypersignal. The functions and algorithms used by Hypersignal--either in user mode, or when used as a DSP engine with the DSPower series software--can be used in intact, or modified to fit application-specific situations, while still maintaining standard Hypersignal-Macro and/or DSPower series sofware functionality.

A project or product can be partially or wholly designed around the basic real-time interrupt structure and DSP and math source code routines available, allowing real-time development directly in the PC environment. By incorporating the real-time disk and video accessibility built-in to the source code interfaces, and using the built-in Hypersignal download function (which accept DSP manufacturer linker output directly), your problem reduces to simply writing and generating the DSP source code specific to your algorithms. You can rely on Hypersignal to allow you to monitor results in real-time, provide you with device drivers to the analog I/O resources of the DSP/acquisition hardware, provide background and foreground

interrupt processing, and allow your data to be transferred continuously to the host computer disk and/or instrumentation displays. The time saved in not dealing with the intricacies of the DSP/acquisition hardware board-host computer interface, the complexity of continuous, real-time transfer of data over the host computer bus, and the disk and instrumentation display housekeeping will be measured in months.

Currently the Analog Devices ADSP-21xx, AT&T DSP32C and DSP3210, Motorola DSP5600x and DSP96002, NEC 77220, and Texas Instruments TMS320C25, TMS320C3x, TMS320C5x, and TMS320C4x devices are supported by DSP Source Code Interfaces.

DSP Source Code Interfaces: More Detail

The DSP Source Code Interfaces include math and DSP library routines, device drivers for analog resources on supported boards, and real-time, interrupt-driven structures that allow foreground and background processing in conjunction with analog I/O sample interrupts. Without modification, each DSP Source Code Interface forms the basis for standard Hypersignal operation, including instrumentation displays, continuous transfer to/from disk, and file-based acceleration. In other words, if you were to assemble and link the standard DSP

source code for each supported DSP chip, without any changes, you would end up with the executable files that are sent with every Hypersignal real-time software package. However, should you then modify, add, or otherwise customize the source code, you could create new executable files that would customize your system's operation.

Changes can be simple, such as interfacing to new analog converters or accessing other resources on the board, or extensive, such as simulating and testing complete algorithms or products. Because the mechanism for transferring data continuously and in real-time between high level Hypersignal functions and the DSP is transparent, you are free to concentrate on the DSP problem at hand. By simply placing the data you need to see or monitor in the correct buffer, at such time or intervals as it becomes ready, you can completely control the content and operation of high level Hypersignal instrumentation and disk transfer functions. Although not necessary, maintaining the Hypersignal infrastructure for transferring data and communicating control parameters offers many advantages throughout the course of your product development. Examples include comparing input data and algorithm output data in real-time, leaving "field-test" capability embedded in your product, and evaluating real-time limits.

For each DSP chip, the built-in Hypersignal download functions accept direct linker output from the manufacturer's development tools. The process of making source code changes and sending those changes into real-time action is a short one:

1) make source code changes using editor
2) compile and/or assemble
3) link (creating new executable file)
4) run Hypersignal

This process is fully documented for each supported DSP device family in Signalogic DSP Software Users Guide chapter 5, Using DSP Source Code Interface Software. Also, DSP source code files are well-commented and clearly point out variables, structures, and procedures related to the intrinsic Hypersignal high level interface. Additionally, the Hypersignal-Macro Series Hardware Reference Guide includes detailed and extensive information on individual DSP/acquisition hardware types supported.

Related Topics

DSP Source Code Interfaces: More Detail

DSP Source Code Interface Partial Function List

DSP Source Code Interface Partial Function List

Below is a partial list of typical source code and utility files

included with each DSP Source Code Interface:

variable and constant definitions (xxx refers to DSP chip family)

DSP chip initialization code, analog I/O and converter initialization code, ISR vector initialization, variable initialization, and initialization of high level interface (xxx refers to DSP chip)

linker specification and map control file (xxx refers to DSP chip)

batch file to assemble typical assembly language source code files

batch file to link typical configuration

various ISRs (interrupt service routines) for different analog I/O functions; each ISR contains different

entry points for specific DSP/acquisition hardware that isolate hardware-dependent analog I/O code and which converge into common, highly optimized code

2-quadrant arctangent approximation

convolution/correlation

fractional and integer divide for fixed-point DSPs; floating-point divide for floating-point DSPs

base 2 exponentiation

complex input, decimation-in-time FFT/inverse FFT

FIR filter implementation

IIR filter implementation (modified direct

form using 2nd-order, or "biquad cascade" sections)

IIR filter implementation (balanced pole-zero form using 2nd-order, or "biquad cascade" sections)

base 2 log approximation

complex input magnitude

location of maximum value in buffer

location of minimum value in buffer

complex input 4-quadrant phase approximation

squareroot approximation

time domain window application and zero-filling

complex-ratio transfer function

calculation

Related Topics

DSP Source Code Interfaces: More Detail

DSP Source Code Interface Partial Function List

1.2.4 DSPower-Block Diagram Software Summary

DSPower® Block-Diagram is a Windows program with block-diagram and graphical control user interfaces; depending on the "DSP Engine(s)" with which it is used, DSPower offers math/DSP simulation and real-time execution of block-diagrams, interactive instrumentation, measurement, displays, visualization, and analysis, data acquisition, and source code generation.

1.2.5 DSPower-C/C++ Code Generator Software Summary

This section not available yet.

1.2.6 Real-Time Convolver Software Summary

The Real-Time Convolver™ is an add-on program intended for use with Hypersignal-Acoustic. It offers FFT-based real-time convolution of analog input data, or accelerated processing of file data, using impulse responses from various sources.

The Real-Time Convolver allows long impulse responses (filters) to be applied to sound inputs in real-time. Depending on the DSP/analog hardware used, real-time sampling rates range from 8 kHz to 48 kHz. A mode is also available that processes sound files. The Real-Time Convolver allows either mono or stereo operation with one or two different filters. Filters can be supplied from outside sources or created with Hypersignal-Acoustic's built-in Difference Equations and FIR/IIR Filter Design functions.

Typical applications include 3-D sound, various real-time sound simulators, precise real-time equalization, and impulse responses gathered from on-site or scale-model acoustic measurements or created by architectural "imaging" programs.

Currently, the Real-Time Convolver can be obtained from Signalogic packaged with a wide range of off-the-shelf DSP/analog

hardware, including Ariel, CAC, National Instruments, and Signalogic boards. When combined with Signalogic's convolution algorithms, these boards offer performance comparable to dedicated convolution instruments, but at a fraction of the cost. For example, with the National Instruments AT-DSP2200, which contains the 32-bit floating-point AT&T DSP32C device, two channels of analog I/O with 92 dB typical dynamic range, and 384k x 32 of zero wait-state SRAM, a 10,000 pt. filter can be applied in real-time to stereo sound data at 44.1 kHz. At lower rates, or with mono operation, filter lengths up to 25,000 pts. are possible. In all cases, a higher amount of DSP chip external memory (SRAM or DRAM) on the board translates to longer filter lengths.

The Real-Time Convolver package requires Hypersignal-Acoustic software, a supported DSP/analog hardware type, DOS 5.0 or higher, and at least a 33 MHz 80386 machine with 4 Mb or more of memory.

1.3 System Requirements

System requirements for Signalogic software components are listed below. If differences exist between software components, requirements may be listed by product name. For more detail on Hypersignal-Macro series software, see

Signalogic DSP Software System Guide section 1.2.1, Hypersignal-Macro Series Software Summary; for more detail on DSPower-HWLib software, see Signalogic DSP Software System Guide section 1.2.2, DSPower-HWLib Software Summary; for more detail on DSPower-Block Diagram software, see Signalogic DSP Software System Guide section 1.2.4, DSPower-Block Diagram Software Summary.

System Requirements Topics

Operating System Requirements

Hypersignal-Macro, Hypersignal-Macro NRT

DOS version 3.3 or higher

Hypersignal-Macro EX, Hypersignal-Acoustic

DOS version 5.0 or higher

All DSPower series packages

DOS version 5.0 or higher
Windows version 3.1 or higher

Additional System Requirements Topics

Operating System Requirements

Processor Requirements

Memory (RAM) Requirements

Video / Graphics Requirements

Mouse Requirements

Math Coprocessor Requirements
Printer Requirements
DSP Engine Requirements
DSP/Analog Hardware Requirements
Windows C/C++ Development Tool Requirements
DSP Development Tool Requirements

Processor Requirements

Hypersignal-Macro series

33 MHz 80386 or faster

DSPower-Block Diagram

33 MHz 80486 (66 MHz or faster is strongly recommended)

Additional System Requirements Topics

Operating System Requirements
Processor Requirements
Memory (RAM) Requirements
Video / Graphics Requirements
Mouse Requirements
Math Coprocessor Requirements
Printer Requirements
DSP Engine Requirements
DSP/Analog Hardware Requirements
Windows C/C++ Development Tool Requirements
DSP Development Tool Requirements

Memory (RAM) Requirements

Hypersignal-Macro, Hypersignal-Macro NRT

448 kbyte free conventional memory needed, 1 Mbyte expanded memory (EMM) needed for publication-quality print

Hypersignal-Macro EX, Hypersignal-Acoustic

same as Hypersignal-Macro, but 4 Mbyte system RAM needed for long FFT, long convolution, long correlation, other macro functions

DSPower-Block Diagram

4 Mbyte, 8 Mbyte strongly recommended

Additional System Requirements Topics

Operating System Requirements

Processor Requirements

Memory (RAM) Requirements

Video / Graphics Requirements

Mouse Requirements

Math Coprocessor Requirements

Printer Requirements

DSP Engine Requirements

DSP/Analog Hardware Requirements

Windows C/C++ Development Tool Requirements

DSP Development Tool Requirements

Video / Graphics Requirements

Hypersignal-Macro series

Any of over 100 different types of hardware supported, including CGA, EGA, VGA, Hercules,

AT&T, and popular 800x600 and 1024x768 formats. If you are unsure of selection to make in ***Graphics/Mouse Config*** menu, try following: 42 (VGA), 65 (popular 800x600), 76 (popular 800x600), 95 (8514 VESA compatible).

DSPower series

Any video/display hardware currently installed under Program Manager ***Main*** group, ***Windows Setup*** item (see "Change System Settings" for list). A 24-bit or 32-bit video board capable of fast, hardware-based bitmap updating is strongly recommended. See Signalogic DSP Software System Guide section 1.4.2, Video and Display Considerations, for more information.

Additional System Requirements Topics

Operating System Requirements

Processor Requirements

Memory (RAM) Requirements

Video / Graphics Requirements

Mouse Requirements

Math Coprocessor Requirements

Printer Requirements

DSP Engine Requirements

DSP/Analog Hardware Requirements

Windows C/C++ Development Tool Requirements

DSP Development Tool Requirements

Mouse Requirements

Hypersignal-Macro series

Interface to memory-resident mouse driver supported; some mice supported directly through serial ports.

DSPower series

Any currently installed inside the Program Manager | Main | Control Panel | Mouse dialog box.

Additional System Requirements Topics

Operating System Requirements

Processor Requirements

Memory (RAM) Requirements

Video / Graphics Requirements

Mouse Requirements

Math Coprocessor Requirements

Printer Requirements

DSP Engine Requirements

DSP/Analog Hardware Requirements

Windows C/C++ Development Tool Requirements

DSP Development Tool Requirements

Math Coprocessor Requirements

All packages

80387 math coprocessor automatically detected and utilized if installed in system. 80486 and higher always used.

Additional System Requirements Topics

Operating System Requirements
Processor Requirements
Memory (RAM) Requirements
Video / Graphics Requirements
Mouse Requirements
Math Coprocessor Requirements
Printer Requirements
DSP Engine Requirements
DSP/Analog Hardware Requirements
Windows C/C++ Development Tool Requirements
DSP Development Tool Requirements

Printer Requirements

Hypersignal-Macro series

Screen-dump and publication-quality print to over 185 printers supported. An appropriate amount of expanded memory (minimum 1 Mbyte) is required in the system to allow publication-quality printing of waveform displays.

DSPower series

Any currently installed inside the Program Manager | Main | Control Panel | Printer dialog box.

Additional System Requirements Topics

Operating System Requirements
Processor Requirements
Memory (RAM) Requirements

Video / Graphics Requirements
Mouse Requirements
Math Coprocessor Requirements
Printer Requirements
DSP Engine Requirements
DSP/Analog Hardware Requirements
Windows C/C++ Development Tool Requirements
DSP Development Tool Requirements

DSP Engine Requirements

For information about DSP Engines, see Signalogic DSP Software System Guide section 1.1, Relationship of Signalogic DSP Software Components.

DSPower-Block Diagram

Hypersignal-Macro, -Macro EX, or -Acoustic v4.0 or higher; MATLAB v4.0 or higher

DSPower-HWLib DLL

Hypersignal-Macro, -Macro EX, or -Acoustic v4.0 or higher; mmsystem.dll (for Windows 3.1 or higher)

Additional System Requirements Topics

Operating System Requirements
Processor Requirements
Memory (RAM) Requirements
Video / Graphics Requirements
Mouse Requirements
Math Coprocessor Requirements

Printer Requirements
DSP Engine Requirements
DSP/Analog Hardware Requirements
Windows C/C++ Development Tool Requirements
DSP Development Tool Requirements

DSP/Analog Hardware Requirements

A supported DSP/analog or DSP/acquisition hardware type (e.g. a plug-in ISA board) is required in the following two (2) cases:

1) When a Hypersignal-Macro, Hypersignal-Macro EX, or Hypersignal-Acoustic package is to be used.

2) When a DSPower-Block Diagram or DSPower-HWLib package is to be used in conjunction with a Hypersignal package as the DSP Engine.

See section 1.4.3, DSP/Analog Hardware Considerations, for more information.

Additional System Requirements Topics

Operating System Requirements
Processor Requirements
Memory (RAM) Requirements
Video / Graphics Requirements
Mouse Requirements
Math Coprocessor Requirements
Printer Requirements

DSP Engine Requirements
DSP/Analog Hardware Requirements
Windows C/C++ Development Tool Requirements
DSP Development Tool Requirements

Windows C/C++ Development Tool Requirements

A Windows C/C++ compiler is currently necessary in two (2) cases:

1) If you have the DSPower-HWLib or DSPower-SPLib packages, which contains C/C++ source code examples. These examples illustrate use of the DSPower-HWLib and DSPower-SPLib DLLs by making calls to DLL functions.

2) If you intend to use the C source code blocks included with DSPower-Block Diagram. Examples include the C source code version of the Difference Equations block.

In these cases, the following Windows C/C++ compilers are suggested:

Borland C/C++ v3.1 or higher
Microsoft Visual C/C++ v1.5 or higher
WATCOM C/C++^{32} v9.5a or higher

Additional System Requirements Topics

Operating System Requirements
Processor Requirements

Memory (RAM) Requirements
Video / Graphics Requirements
Mouse Requirements
Math Coprocessor Requirements
Printer Requirements
DSP Engine Requirements
DSP/Analog Hardware Requirements
Windows C/C++ Development Tool Requirements
DSP Development Tool Requirements

DSP Development Tool Requirements

TMS320C25 Source Code Interface

Texas Instruments "DSP COFF Assembler", v5.05 or higher, Texas Instruments "DSP COFF Linker", v5.05 or higher

DSP5600x Source Code Interface

Motorola "DSP56000 Assembler", v4.1 or higher, Motorola "DSP Linker", v4.1 or higher

DSP32C Source Code Interface

AT&T v1.3.1 "d3as Assembler", v1.3.1 or higher, "d3ld Linker", v1.3.1 or higher

TMS320C3x Source Code Interface

Texas Instruments "TMS320C3x/C4x COFF Assembler", v4.5 or higher, Texas Instruments "TMS320C3x/C4x COFF Linker", v4.5 or higher

ADSP-21xx Source Code Interface

Analog Devices "ASM21" assembler, v1.13 or higher, Analog Devices "LNK21" linker, v1.12 or higher

DSP32C Source Code Interface

AT&T v1.3.1 "d3as.exe" assembler, v1.3.1 "d3ld.exe" linker

DSP32C Source Code Interface

AT&T v1.3.1 "d3as.exe" assembler, v1.3.1 "d3ld.exe" linker

DSP32C Source Code Interface

AT&T v1.3.1 "d3as.exe" assembler, v1.3.1 "d3ld.exe" linker

DSP32C Source Code Interface

AT&T v1.3.1 "d3as.exe" assembler, v1.3.1 "d3ld.exe" linker

Additional System Requirements Topics

Operating System Requirements

Processor Requirements

Memory (RAM) Requirements

Video / Graphics Requirements

Mouse Requirements

Math Coprocessor Requirements

Printer Requirements

DSP Engine Requirements

DSP/Analog Hardware Requirements

Windows C/C++ Development Tool Requirements

DSP Development Tool Requirements

1.4 System Considerations

Signalogic DSP Software packages have been designed to run on IBM 386, 486, and Pentium machines using Microsoft Windows v3.1 or higher. Hypersignal-Macro series software packages have also been designed to run stand-alone under DOS v3.3 or higher with over 100 different types of graphics hardware and several different types of mice installed.

Signalogic DSP Software packages will likely function correctly on many "IBM-compatible" or similar machines with compatible or similar graphics hardware. However, the word "compatible" means something different to every hardware and software manufacturer, and Signalogic can make no guarantee concerning compatibility.

One area that has a widespread reputation for being a source of problems is video drivers. See Signalogic DSP Software System Guide section 1.4.3, Video Considerations, for more information.

1.4.1 Memory Considerations

[@j conventional memory, virtual memory, DOS extender]

This section covers memory considerations of all types, including conventional memory,

virtual memory, differences when used in protected-mode vs. real-mode, and typical DSP-oriented function use of available memory. Both Hypersignal-Macro series software and DSPower series software issues are covered.

Hypersignal Memory Considerations

The standard Hypersignal core functions exclusively utilize "conventional DOS" memory, and use overlays to virtualize this area for both program and data information. Some of the new processing functions added to Hypersignal-Macro EX and Hypersignal-Acoustic have been implemented as protected-mode programs controlled by macro functions. Many of these functions require large amounts of program and data memory. Examples include long FFT, convolution, and correlation, MLS stimulus & response, etc. You should have at least 8 Mbyte of extended memory in your system if you want to use these functions "comfortably", without causing virtual memory disk swaps.

The DOS extender program DOS4GW.EXE is used to load and run the protected-mode programs. Once running, the protected-mode programs request extended memory based on processing needs. Some important notes about amount of extended memory and its usage:

1) The amount of memory installed in your machine is not the amount available. The first 1 Mbyte is devoted to "conventional memory" and housekeeping machine requirements. Also, other programs use extended memory, such as SMARTDRV.EXE, which is commonly installed with later versions of DOS and Windows, and which takes up 2 Mbyte of extended memory in its default mode. Finally, expanded memory may take a chunk, too--although expanded memory is used less often with state-of-the-art systems.

To determine if your system is configured to use some portion of extended memory as the older "expanded" type of memory, look for the line "device=EMM386.EXE ..." in your system's CONFIG.SYS file. If this line exists, but does not contain the attribute "NOEMS", then it some amount of expanded memory may be configured. See DOS or Windows manuals for more information.

Thus, a machine with only 4 Mbyte of extended memory often ends up with not much available for user programs. You can run the PMINFO.EXE program included with Hypersignal to determine the amount of extended memory actually available.

2) The protected-mode programs can be run under Windows (i.e. if Hypersignal has been launched directly from Windows, or run

from a DOS command prompt that was launched from Windows). When run under Windows, the protected-mode program's use of system memory is fully compatible with Windows and whatever Windows applications are already running, including access to Windows virtual memory, which extends the amount of memory available on a system by swapping unused sections to disk file. ***However, do not expect virtual memory to help with typical DSP programs.*** For example, lcorr.exe, lconv.exe, and lfft.exe, are exceedingly slow when they use virtual memory, because the bit-reversed memory-access nature of the FFT routine used by these programs causes excessive "disk thrashing". Disk thrashing occurs when virtual memory is being swapped in and out from disk file so frequently that programs that should take seconds end up taking hours. To run larger FFT sizes, larger filter sizes, and larger MLS stimulus & response sizes, it is strongly suggested that you upgrade (expand) the amount of memory in your computer. For example, on a typical 486 system running Windows 3.1, and with SMARTDRV.EXE installed in the AUTOEXEC.BAT file and no EMS (expanded) memory installed, 8 Mbyte of extended memory typically would allow a 262,144 pt. complex FFT in the lfft.mac function without disk thrashing. But, using the lcorr.mac or lconv.mac functions with an

implied FFT size of 262,144 (i.e. a filter size of 200,000 pts.) would require a full 16 Mbyte, due to the extra memory needed for convolution and correlation.

A note about DOS versions less than 5.0: DOS4GW.EXE still needs some conventional memory to get started--before loading a protected-mode program--and this can be a problem with DOS versions less than 5.0. In these cases, the combination of Hypersignal, the MET_SHEL.EXE graphics driver, DOS's command.com, and other utilities can occupy enough conventional memory to keep DOS4GW.EXE from running. When this happens, you will see "not enough memory" error messages when attempting to run the extended macro functions. Here are some suggestions to deal with this problem:

- use DOS 5.0 or higher, and make sure the DOS=HIGH,UMB command is in your CONFIG.SYS file
- run PMINFO.EXE and RMINFO.EXE, included with DOS4GW.EXE, to diagnose the current memory state and protected-mode compatibility of your machine, and make sure there's not a basic problem with switching into protected mode
- start "trimming" your system: REM-out CONFIG.SYS and AUTOEXEC.BAT command lines such as FASTOPEN and SMARTDRV, and remove other memory-resident utilities that take up conventional memory

Note that even for DOS 3.3x through 4.0x this is not a problem when running Hypersignal software in engine mode, because the MET_SHEL.EXE graphics driver is not loaded in this case, thus freeing up an additional 120k of

conventional memory. Note: DSPower-Block Diagram software invokes Hypersignal-Macro series software packages in engine mode when executing Hypersignal macro language blocks.

Memory Requirements Topics

Hypersignal Memory Considerations

DSPower Memory Considerations

DSPower Memory Considerations

As with most Windows applications, memory is the critical issue for DSPower-series software performance. Below are some system memory recommendations and comments regarding use of DSPower-series software:

- install at least 8 Mbyte of system memory (see also section 1.4.1, Memory Considerations)
- avoid active use of known memory hog applications at the same time as DSPower-Block Diagram; for example, Microsoft Word
- *do not* set up your system for virtual memory beyond the recommended maximum size, and do not enable 32-bit disk access (see "386 Enhanced" icon in Program Manager | Main | Control Panel). Both of these settings can decrease rather than increase performance. Especially in the case of DSP or math functions using very large operation sizes (e.g. 262,144 pt. FFT size), extra virtual memory can shut a system down, because the math or DSP function will attempt to use this memory as it would system memory. Examples include the lfft.mac Hypersignal protected-mode DOS program, and DSPower-Block Diagram wavelet transform function.

Memory Requirements Topics

1.4.2 Performance Considerations

This section provides some suggestions for achieving and maintaining maximum system performance when using Hypersignal-Macro series and/or DSPower series software. For additional information , see also chapter 5, Troubleshooting.

Hypersignal Performance Considerations

[@j DOS box, protected-mode, real-mode, .pif files]

The highest possible performance for Hypersignal packages is made possible when they are run from a DOS prompt, *and Windows is not installed*. In this mode, called real-mode, Hypersignal installs its own mouse, video and printer drivers; the only possible problems are hardware incompatibility or not enough available conventional DOS memory. If a Hypersignal package fails to load from a DOS prompt, try removing less-than-absolutely-necessary utilities and memory resident programs from your CONFIG.SYS and AUTOEXEC.BAT files, one by one.

If Hypersignal is executed from Windows (either from a DOS

prompt started from Windows, or using the Hypersignal icons included with the software), then it will be running in a "DOS box", which simulates a stand-alone DOS environment. This is only a simulation; the host machine is actually not in real-mode, but in protected mode, which is only possible with 386 and 486 processors. When a DOS program is executed under Windows, "time-slices" are always allocated to Windows, even though Windows is in the background (not visible). Thus, the potential exists for Hypersignal, and all its functions, displays, and hardware-based instrumentation, to run slower than if truly stand-alone (in real-mode). To minimize slowdown when running Hypersignal under Windows, take the following steps:

- install Hypersignal-Macro or Hypersignal-Acoustic with an icon in Program Manager (see Runing Hypersignal in User Mode Under Windows, in section 3.2.5, Installing and Configuring Hypersignal for User Mode Operation). This will cause automatic use of hs.pif, hsm.pif, or hsa.pif files included on the Hypersignal distribution diskettes which are optimized for high performance under Windows.

- decrease time-slices given to Windows when in background mode to the minimum value of 1. To do this, select Program Manager | Main | Control Panel | 386 Enhanced.

Even with the above measures in place, you can still expect an approximate 10% slowdown in Hypersignal performance when running under Windows (in protected-mode) as opposed to stand-alone (real-mode). This is the nature of protected mode, and is true for all DOS programs.

NOTE: When installing Hypersignal with the Signalogic Software Setup Utility, which is included with DSPower, the above steps and performance increasing measures, including installation of Hypersignal icons under Windows, are performed automatically.

The Hypersignal .pif files mentioned above are set to optimize performance of Hypersignal. However, if you wish to create your own .pif files, the following settings are recommended:

- operation set to full-screen
- emulate text mode
- start in text mode (video memory)
- desire at least 384 kbyte of memory (maximum used is 640k)
- use high memory
- close window on exit
- no XMS, EMS, or locked memory is needed
- priority and background vs. exclusive operation are not important
- detect Idle Mode should be turned off (VERY IMPORTANT FOR PERFORMANCE)

NOTE: When using Hypersignal with DSPower-Block Diagram, .DSPower uses .pif settings from the files "hsmacro.pif" and "hsacous.pif" located on the DSPower subdirectory. These files should not be edited or changed under any circumstances.

Performance Considerations Topics

Hypersignal Performance Considerations

DSPower Performance Considerations

DSP/Analog Hardware Performance Considerations

DSPower Performance Considerations

DSPower-series software performance issues can be divided into three categories:

System memory.

System video display.

System disk access.

Memory is the most critical issue: too small an amount installed, or situations where other active applications have left too little memory available, can bring any Windows application to its knees, including DSPower-series software. Video drivers and display boards, if not optimized and accelerated, can be a source of poor performance, especially in the case of a graphically-oriented application such as DSPower-Block Diagram. The same holds true for disk access, which is important to DSPower-Block Diagram because of its intensive and constant manipulation of source code files. Currently, the following recommendations are given:

- install at least 8 Mbyte of system memory (see also section 1.4.1, Memory Considerations)

- avoid active use of known memory hog applications at the same time as DSPower-Block Diagram; for example, Microsoft Word
- *do not* set up your system for virtual memory beyond the recommended maximum size, and *do not* enable 32-bit disk access (see "386 Enhanced" icon in Program Manager | Main | Control Panel). Both of these settings can decrease rather than increase performance. Especially in the case of DSP or math functions using very large operation sizes (e.g. 262,144 pt. FFT size), extra virtual memory can shut a system down, because the math or DSP function will attempt to use this memory as it would system memory. Examples include the lfft.mac Hypersignal protected-mode DOS program, and DSPower-Block Diagram wavelet transform function.
- install a video accelerator board with 24-bit or 32-bit access over a local bus, with as much hardware-based support for typical video operations (bitmap painting, line operations, etc.) as possible
- install a disk drive controller with local bus access

Performance Considerations Topics

Hypersignal Performance Considerations

DSPower Performance Considerations

DSP/Analog Hardware Performance Considerations

DSP/Analog Hardware Performance Considerations

One of the most common areas of concern when using DSP/analog hardware is maximum transfer rates to and from hard disk. To achieve maximum performance, here are some pointers and suggestions:

A fast system. Nothing helps like raw talent. A fast processor (for

example, a 66 Mhz 486 or better), a fast hard disk, preferably with its controller card located on a VESA local bus or PCI bus, video controller card located on a VESA local bus or PCI bus, and an internal bus speed of at least 8.33 MHz, and possibly higher, will make a big difference. Locating the hard disk controller and video controller cards on a local bus is important, because the DSP/analog hardware should then have exclusive use of the standard ISA bus during acquisition (99% of off-the-shelf DSP/analog hardware are ISA bus plug-in cards).

Some network adaptor cards and other peripheral cards can actually slow down a host PC's internal bus (ISA bus) to rates as low as 4.77 MHz because of the way they respond to bus acknowledgement signals. If you suspect your DSP/analog hardware disk performance is

too slow, you might try temporarily taking out all other peripheral boards in your system, and re-measuring.

Make sure your available hard disk space is not fragmented. Non-contiguous file space can cause delays in continuous acquisition to disk, which could translate into gaps in the data. You can use an application like Norton Desktop to periodically run a disk-defragmenter utility, say once a week. This should ensure large chunks of free, contiguous disk space at any given time. Also, if the software being used allows it, you might try increasing the number of buffers being used; more buffers act as a greater "shock absorber" for delays caused by non-contiguous file space or bad-tracks.

Do not use disk space compression; e.g. DoubleSpace, Stacker, etc. These utilities take extra

time to compress data before it is actually stored to disk. Normally, this time is not noticed because a big memory cache is enabled for the system's disk I/O, and data is written first to the cache, allowing programs to think the transfer is complete. But, when long data acquisition operations take place, the cache quickly gets full, and data must then be stored to disk at the same rate as it is coming in. At this point, the performance degradation of a compression utility will be quite noticeable.

For some systems, continuous disk transfer performance is actually increased if 32-bit disk access is disabled. This option is located under Program Manager | Main | 386 Enhanced | Virtual Memory.

For some systems, continuous disk transfer performance is actually increased if the SMARTDRV.EXE

utility is not loaded. This option is usually specified inside the system's AUTOEXEC.BAT file, and sometimes in its CONFIG.SYS file (in the form "device = C:\WINDOWS\SMARTDRV.EXE" or similar). The reason? Because long, continuous disk acquisition operations at high sampling rates eventually fill up the internal memory cache used by SMARTDRV.EXE; afterwards, the advantages of caching are lost as data must leave the cache at same rate it enters. At this point, the overhead of cache processing does not contribute to disk performance, and may detract. Note, however, that disabling SMARTDRV.EXE will have a detrimental effect on the performance of your system overall, especially for word processing, database, communications, and other applications that make extensive random file accesses.

If disabling SMARTDRV.EXE turns out to be necessary for the DSP/analog hardware in your system to achieve continuous acquisition at a certain required maximum sampling rate, you should wrap the disabling into a macro program or DSP engine function, so that it can be disabled and re-enabled automatically.

You can use the "sliding window" FFT technique to check for discontinous data in a recorded waveform file that you suspect as possibly having gaps. To do this, perform an FFT function (in Hypersignal-Macro series software, use FFT Generation, in DSPower-Block Diagram use the FFTEx function) with the following parameters:

Framesize of 250
FFT size of 512 (FFT order = 9)
Overlap of 50%
Hamming Window

This will cause the input waveform to be analyzed with a sliding window that will highlight sharp transitions such as likely to be

associated with a discontinuity. Showing the results in the 2-D spectrograph (contour) display function; can do two things: 1) display the entired waveform at once, and 2) allow a "look down" into the frequency data from 0 to -70 dB or -80 dB. If there are gaps in the data they should show up as one or more brightly colored vertical lines, and possibly evenly and/or repetitively spaced. The vertical line occurs because the discontinuity is seen by the FFT analysis as wideband noise; the even and/or repetitive spacing--if present--may represent the internal buffer size used by the acquisition operation.

Performance Considerations Topics

Hypersignal Performance Considerations

DSPower Performance Considerations

DSP/Analog Hardware Performance Considerations

1.4.3 Video and Display Considerations

This section provides some suggestions for achieving and maintaining maximum video and display performance when using Hypersignal-Macro series and/or DSPower series software. For additional information , see also section 4.2, Troubleshooting Video and Display Problems.

Hypersignal Video and Display Considerations

For detailed information, see Hypersignal-Macro Series Users Manual section 2.4, Selecting Graphics and Mouse Drivers.

When Hypersignal packages are run from a DOS prompt, *and Windows is not installed*, Hypersignal installs its own video

drivers. This is the simplest situation possible; video driver selections can be made directly from a Hypersignal Utilities menu selection. When running Hypersignal from Windows (either from Hypersignal icons or from a DOS prompt started from Windows), the situation is more complicated. This is mainly because Windows must temporarily deactivate and save the context of its installed video driver, and allow the DOS program to *perceive* it is running stand-alone. Problems then sometimes appear in transition, when video control is transferred from one driver to another. Display drivers commonly used in the Windows environment sometimes exhibit such transition problems with DOS programs. Such problems include losing "sync" or fuzzy display when going to the DOS mode, or suffering dramatic (and sometimes garish) color changes when returning to the Windows mode. Another observed problem is background "clearing" problems inside Hypersignal, once it is running. See Signalogic DSP Software System Guide section 4.2, Troubleshooting Video and Display Problems, for suggested workarounds to these problems.

Basically, we have found the standard Windows 3.x VGA driver (640x480 resolution) to be reliable, and allow use of Hypersignal either from a DOS shell command line prompt, or from the Hypersignal icons without problems. However,

use of the standard Windows 3.x "SuperVGA" driver, or 800x600 and 1024x768 drivers provided by the graphics board manufacturer has not proven to be so reliable. In general, we have found that the manufacturer's drivers are almost always better than the general Windows driver equivalents. Note that any incompatibility or conflict arising from alternating use of Hypersignal video drivers and installed Windows video drivers is avoided when Hypersignal is used with DSPower; in which case the Hypersignal video drivers are not loaded or used.

Video and Display Considerations Topics

Hypersignal Video and Display Considerations

DSPower-Block Diagram Video and Display Considerations

DSPower-Block Diagram Video and Display Considerations

A 24-bit or 32-bit video board is highly recommended. If you are using a video board that has only the standard AT-style ISA connectors, then you do not have more than a 16-bit video board. Typical 24-bit or 32-bit video boards plug into VESA or PCI local bus connectors, in addition to occupying a standard ISA slot. In almost any Windows application, including DSPower-Block Diagram, user-interface functions are video intensive. In DSPower-Block Diagram, video operations

are further intensified when one or more waveform displays or instruments are onscreen at the same time.

NOTE: In DSPower-Block Diagram, if you see the cursor "flashing" when it is positioned over an updating waveform display, then you do not have a video board capable of fast, hardware-based bitmap painting. It is strongly recommended that you obtain a 24-bit or 32-bit video board capable of hardware-based bitmap updates.

Symbols, dialog boxes, and other Windows graphical items used in DSPower-Block Diagram will appear best at video (screen) resolution of 800x600 or higher (1024x768).At 640x480 resolution (i.e. standard VGA), block symbols can appear "sketchy". You can avoid this by disabling the "Object Size Independent of Display Resolution" option in the Worksheet selection of the Options menu. However, if you do this, then blocks will become bigger at 640x480 resolution, and smaller at 1024x768 resolution.

Likewise, dialog boxes, menus, and toolbar icons can appear disproportionately large at 640x480 resolution, but there is no option controlling the appearance of these items. This discussion does not apply to waveform displays and instruments, which scale in a truly automatic fashion independent of

video resolution (although text labels in these displays and instruments can become unreadable at small window sizes).

From experience with various 386 and 486 clone machines, we highly suggest that you use the SuperVGA (800x600) or 1024x768 driver provided by the manufacturer of your video board. It has been our experience that the generic SuperVGA driver included with Windows 3.x does not always work reliably with some video boards. Thus, you might want to take the time to install SuperVGA drivers using the setup utility provided on the video board manufacturer's disk, if you have not already done so.

Another suggestion is to avoid more than 256 colors, depending on the video driver. Experience has shown that video driver bugs appear more often and in more severe form when the video memory needed for any given operation increases, which is the case with increasing number of colors.

Video and Display Considerations Topics

Hypersignal Video and Display Considerations

DSPower-Block Diagram Video and Display Considerations

1.4.4 System Drivers and Resident Program Considerations

This section deals with those miscellaneous collection of Windows and DOS system utilities, drivers, and resident programs that lurk in the background--most of the time unnoticed, but sometimes the source of unexplained problems, performance decreases, or bizarre system behavior.

SHARE.EXE

SHARE.EXE is a Microsoft resident program that originated with later versions of MS-DOS, but also has a Windows 3.1x VxD equivalent. SHARE.EXE can be used to govern access to files on the basis of the mode in which they were opened, and who "owns" them. These features were not originally part of the MS-DOS system, and so SHARE.EXE is one of those typical "growths" that have quietly sprung up around MS-DOS and end up in widespread, but not always flawless, use. For example, some Windows applications require SHARE.EXE in order to run correctly (e.g. Microsoft Word), yet other applications cannot run with it installed (e.g. Ventura Publisher). Typically, SHARE.EXE is found somewhere in the system's AUTOEXEC.BAT file; there may also be a reference in the "[386Enh]" section of the Windows system.ini file.

Hypersignal-Macro series and DSPower-series software can run with SHARE.EXE either installed or not installed in your system. Both types of software have been tested extensively with and without SHARE.EXE. If installed, DSPower-Block Diagram takes advantage of SHARE.EXE to prevent simultaneous editing on the same waveform file by two different display/editing functions. If for some reasone, you must remove SHARE.EXE from your system at some period during which you are running DSPower-Block Diagram, avoid editing two different copies of the same waveform file at the same time.

System Drivers and Resident Program Considerations Topics

SHARE.EXE
EMM386.EXE

EMM386.EXE

The original use of EMM386.EXE was to provide system control over something called "expanded memory", which provided a system with access to large amounts of memory through a small (64 kbyte) window located in conventional "high", or upper, system memory (between 640k and 1024k). However, as amounts of standard system memory (also referred to as extended memory) increased, expanded memory usage has almost disappeared, because it is

too slow, requires the extra cost of interface circuitry, and is more difficult for software to use.

But! That's not the end of the story. EMM386.EXE continues to play an important role in the typical system, because it can act as gatekeeper for *all* memory located between 640k and 1024k. Thus, peripheral hardware or software components in a system, such as network cards, memory-resident DOS software programs, add-in peripheral cards (such as DSP/analog hardware), etc. end up dealing with EMM386.EXE. For a thorough discussion of how EMM386.EXE affects installation and usage of typical DSP/analog hardware, see "EMM386.EXE Considerations (Reserving Memory Areas)" in section 4.3.2, Memory-Mapped Transfer Gotchas.

System Drivers and Resident Program Considerations

SHARE.EXE

EMM386.EXE

1.4.5 DSP/Analog Hardware Considerations

In cases where a Hypersignal-Macro series package is in use, external drivers provided by the DSP/analog hardware manufacturer or other 3rd party are not required;

Hypersignal handles all hardware communication.

If the DSPower-Block Diagram or DSPower-HWLib packages are not being used with a Hypersignal-Macro series package, but in conjunction with drivers supplied by the DSP/analog hardware manufacturer, then these drivers must be compatible with the Windows Multimedia 3.1 standard, including the mmsystem.dll component shipped with Windows 3.1.

DSP/Analog Hardware Considerations Topics

DSP/Analog Hardware Performance Considerations

Using Hardware Manager

Troubleshooting DSP/Analog Hardware Problems

For more information, see also:

DSP/Analog Hardware Performance Considerations in section 1.4.2, Performance Considerations.

Signalogic DSP Software Users Guide chapter 8, Using Hardware Manager.

System Guide section 4.3, Troubleshooting DSP/Analog Hardware Problems.

1.5 Technical Support

At the present time, after-sale technical support is available free-of-charge, on an unlimited basis, using the following methods. If you have read what there is to read, and consulted the chapter 5, Troubleshooting, section, and your problem still doesn't go away, please try the following:

1. send e-mail to:

DSPhelp@signalogic.com

or

72113.1412@compuserve.com

2. send fax to 214-343-0163

3. if absolute urgency is required, contact Signalogic technical support at 214-343-0069

NOTE: Technical support calls to 1-800-DSPower are not accepted.

2 Documentation

Overview

The Documentation section provides an overall look at Signalogic DSP Software documentation, including organization and conventions used. Additional references and sources are also listed.

Note that the Signalogic DSP Software System Guide, and all DSPower documentation, have both an online Help form and a printed form. These forms are actually created from the same source, and so will appear similar. However, there are significant differences, as listed in section 2.3, Documentation Conventions, below.

Main Topics

Documentation Assumption of Previous Knowledge

Documentation Organization

Documentation Conventions

Documentation References and Additional Sources

2.1 Documentation Assumption of Previous Knowledge

The Signalogic DSP Software documentation assumes you are reasonably familiar with the following general areas:

1. **PC, DOS, and Windows terminology.** Examples include system resources of a PC (such as type and speed of processor, RAM, printer and serial ports, hard disk and CD ROM drivers, video display resolution, etc.), use of the DOS command line prompt, running applications and making menu and dialog box selections under Windows, and configuring a system using the Control Panel and Windows Setup icons under Windows.
2. **DSP terminology and nomenclature.** A sampling of subjects includes data and file acquisition, time and frequency domain analysis, digital filters, FFT, convolution, correlation, wavelet transform, z-transform, discrete sequences, difference equations, real-time processing, simulation, multirate sampling, minimum phase, joint time-frequency display, transfer function, and pole-zero analysis.
3. **DSP/analog hardware terminology.** Examples include sampling rates, multichannel, dual-port memory, instruction cycle time, DSP chip assembly language, downloading executable DSP files, Nyquist rate, antialiasing filters, sigma-delta converters, circular buffers, onchip RAM, and floating-point vs. fixed-point.

Related Documentation Topics

Documentation Assumption of Previous Knowledge

Documentation Organization

Documentation Conventions

Relationship Between Online and Printed Documentation

Online Documentation Conventions

2.2 Documentation Organization

Signalogic DSP Software documentation is currently organized into the following books and online Help files (Help files shown in parentheses):

Signalogic DSP Software System Guide[1] (sysguide.hlp)

Covers system issues for all Signalogic DSP Software, including DSPower series and Hypersignal-Macro series software. Includes relationship and summary of products, system requirements and considerations, installation, documentation, performance issues, and troubleshooting.

Main System Guide Topics

Summary of Software Quick Reference

System Requirements Quick Reference

System Considerations Guide Quick Reference

Documentation Quick Reference

Installation Quick Reference

Troubleshooting Quick Reference

Signalogic DSP Software Users Guide (useguide.hlp)

Provides instructions on using DSPower series software, DSP Source Code Interface software, and compressed instructions on using Hypersignal-Macro series software.

Main Users Guide Topics

Using Hypersignal-Macro Series Software

Using DSP Source Code Interface Software

Using DSPower-HWLib and DSPower-SPLib Software

Using DSPower-Block Diagram Software

Using Engine Manager

Using Hardware Manager

Signalogic DSP Software Reference Guide (refguide.hlp)

Provides detailed technical reference for DSPower functions, DSP Engine functions, and DLL calls. Includes lists of DSP/analog hardware designators and error messages.

Main Reference Guide Topics

Block Reference

Function Reference

DSP/Analog Hardware Reference

Waveform File Formats

Struct Definitions

Error and Status Messages

Signalogic DSP Software Tutorial (tutorial.hlp)

Provides step-by-step examples on using DSPower series software.

Hypersignal-Macro Series Users Manual

Provides fully detailed instructions on using Hypersignal-Macro series software.

Hypersignal-Macro Series Hardware Reference Guide

Provides in-depth description of hardware supported by Hypersignal-Macro series software, including installation and configuration, analog I/O and digital specifications, DSP device and memory issues, and usage with software.

Hypersignal-Macro Series Software Reference Guide

Provides concise description of input fields and commands for Hypersignal-Macro series functions, displays, and instruments.

Hypersignal-Macro Series Tutorial

Provides step-by-step examples of using Hypersignal-Macro series software in user mode.

The online Help file shown in parentheses above can be invoked from icons in the Signalogic DSP Software group in Program Manager, or during the course of DSPower software operation. Note that Hypersignal-Macro series software also has a native online Help facility when operated in user mode.

Notes

[1]This manualhelp file.

Related Documentation Topics

Documentation Assumption of Previous Knowledge

Documentation Organization

Documentation Conventions

Relationship Between Online and Printed Documentation

Online Documentation Conventions

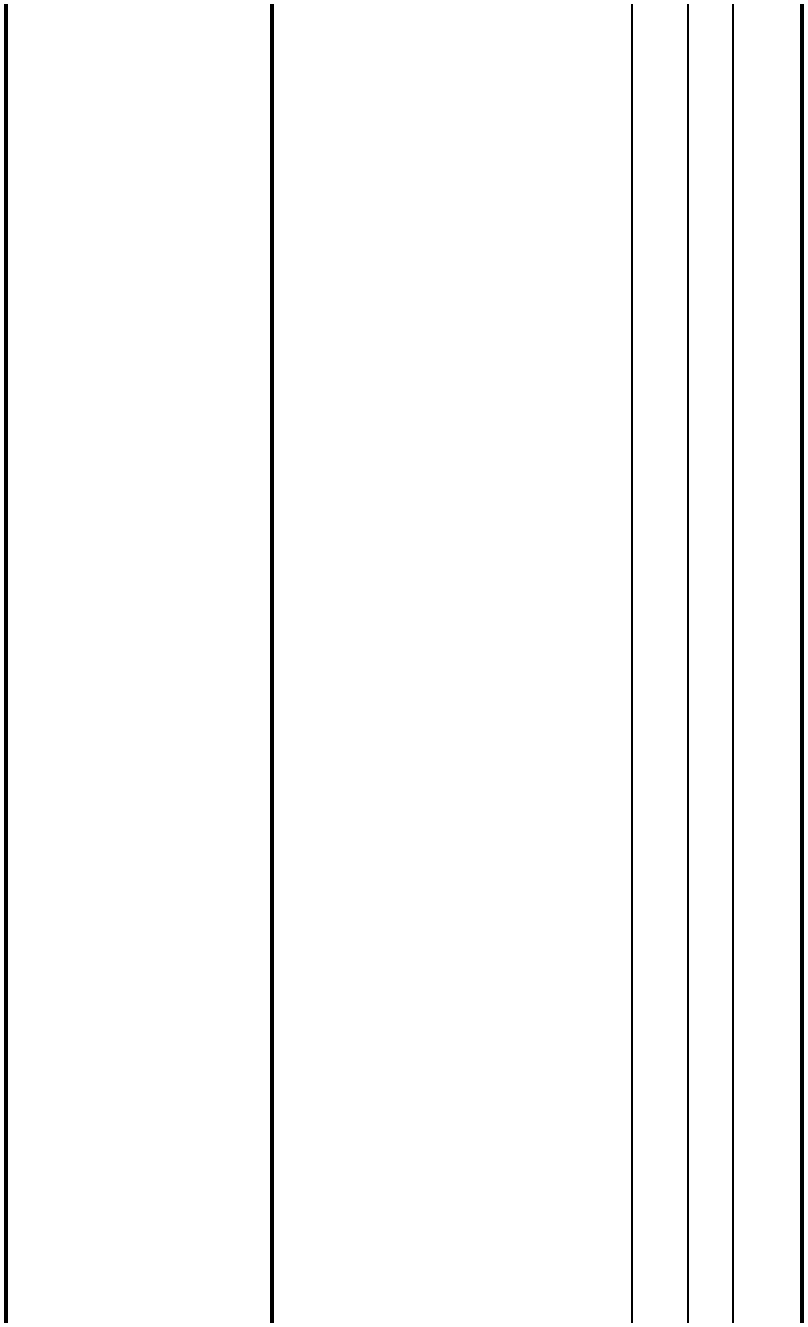
Documentation References and Additional Sources

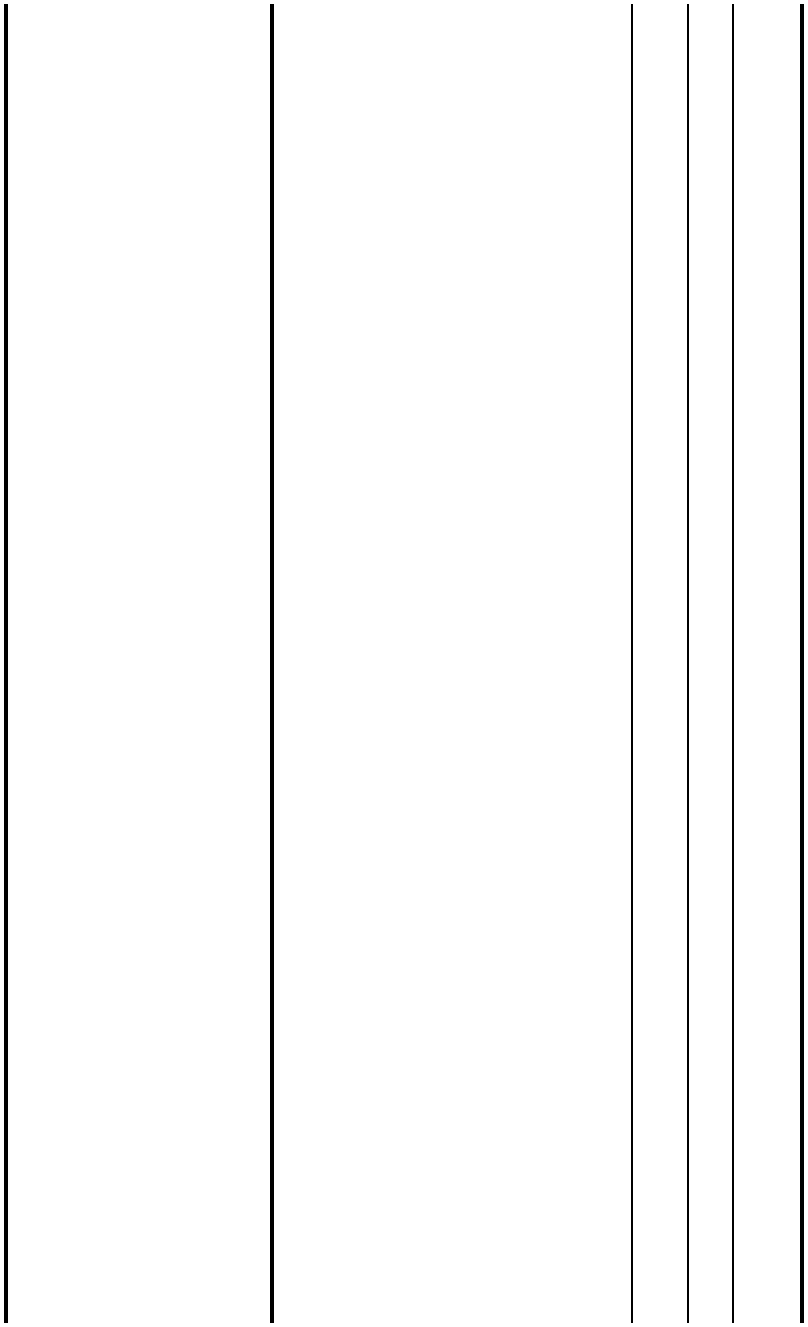
2.3 Documentation Conventions

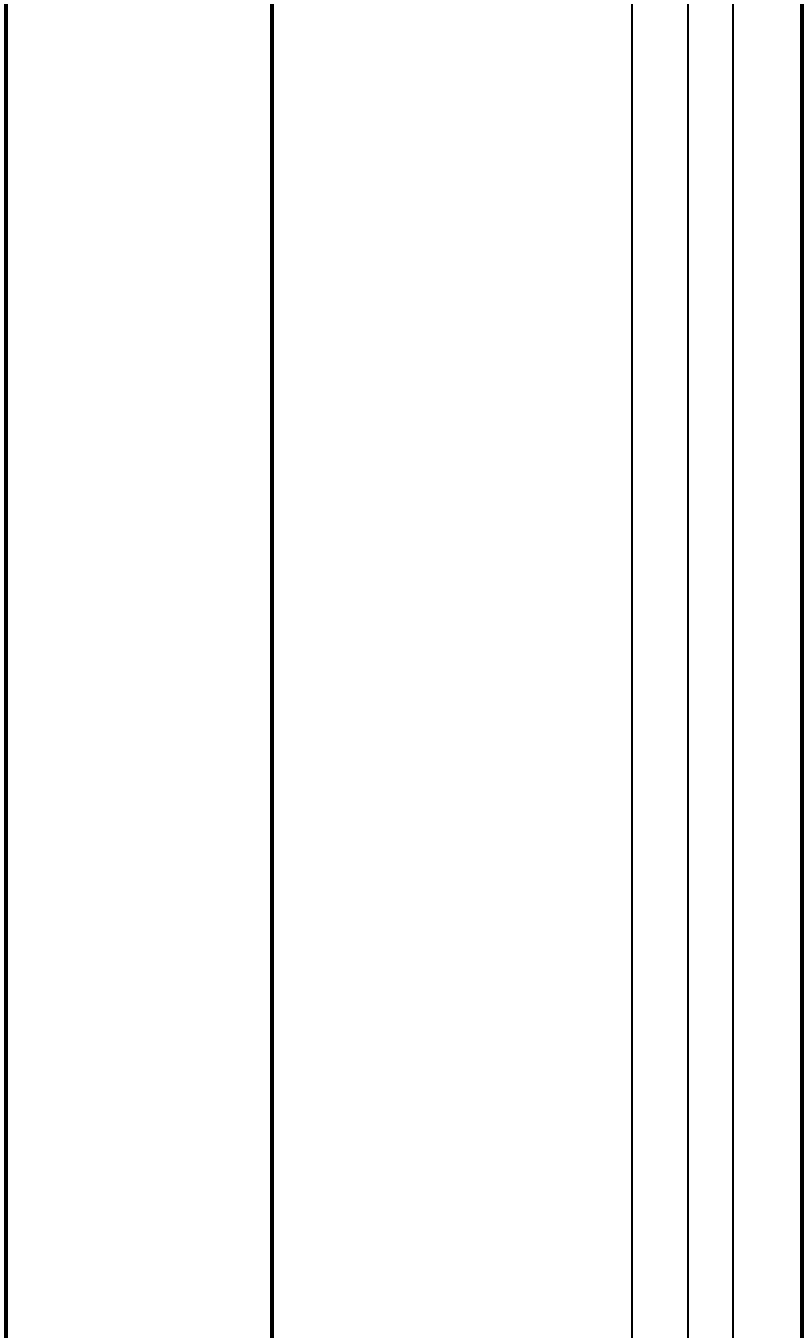
Following are some conventions typically used in Signalogic DSP Software documentation. Note that each convention is marked as to whether it used in DSPower series (table heading "D"), Hypersignal-Macro series (table heading "H"), or DSP Source Code documentation (table heading "SC"), or combination thereof.

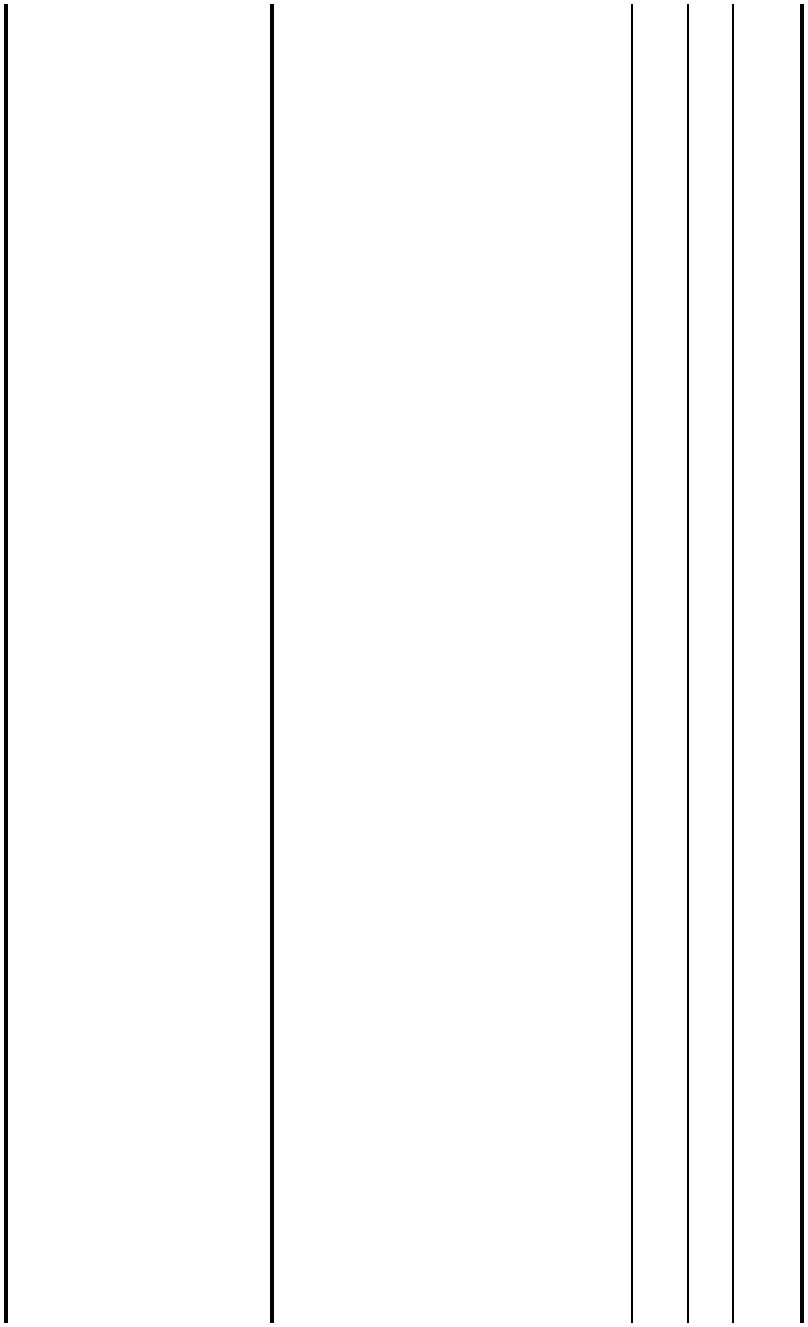
Table 10 Signalogic DSP Software Documentation Conventions

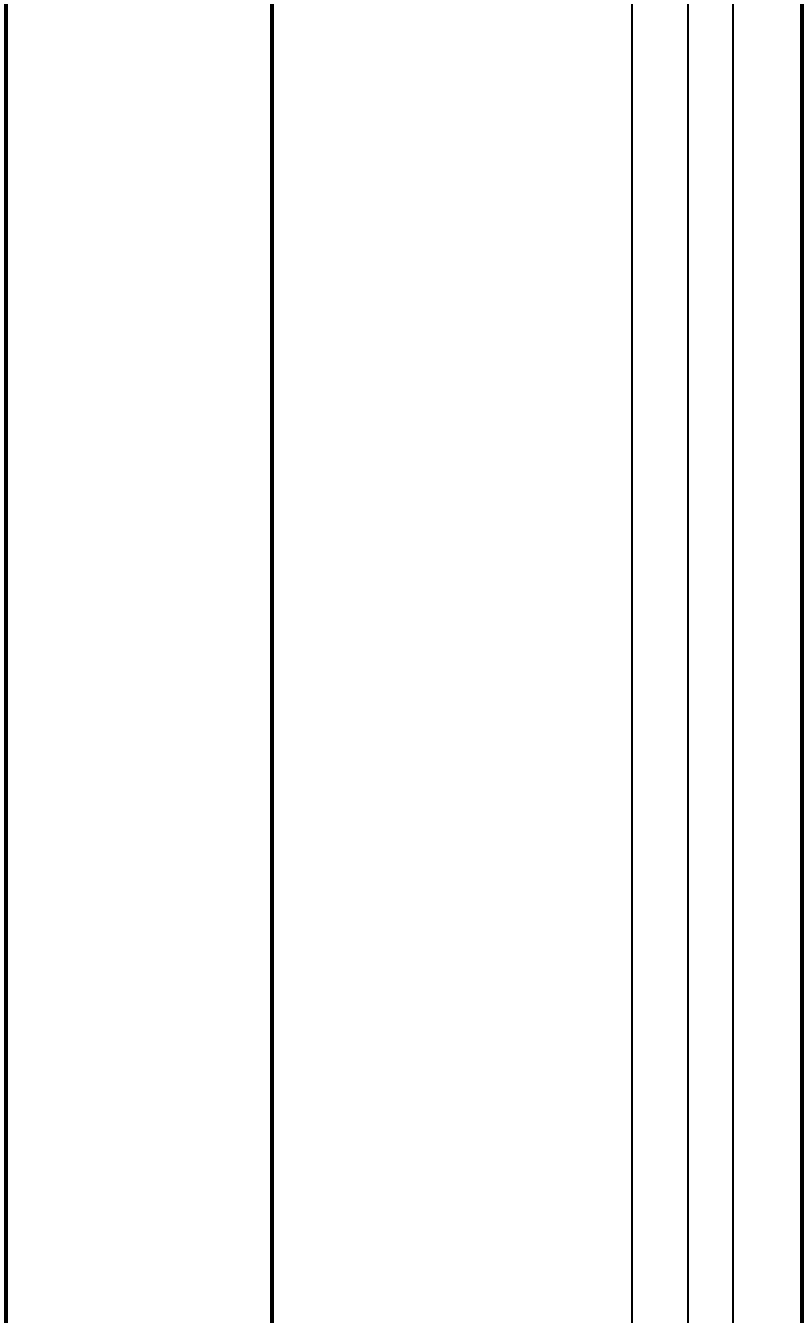
Convention	Meaning			

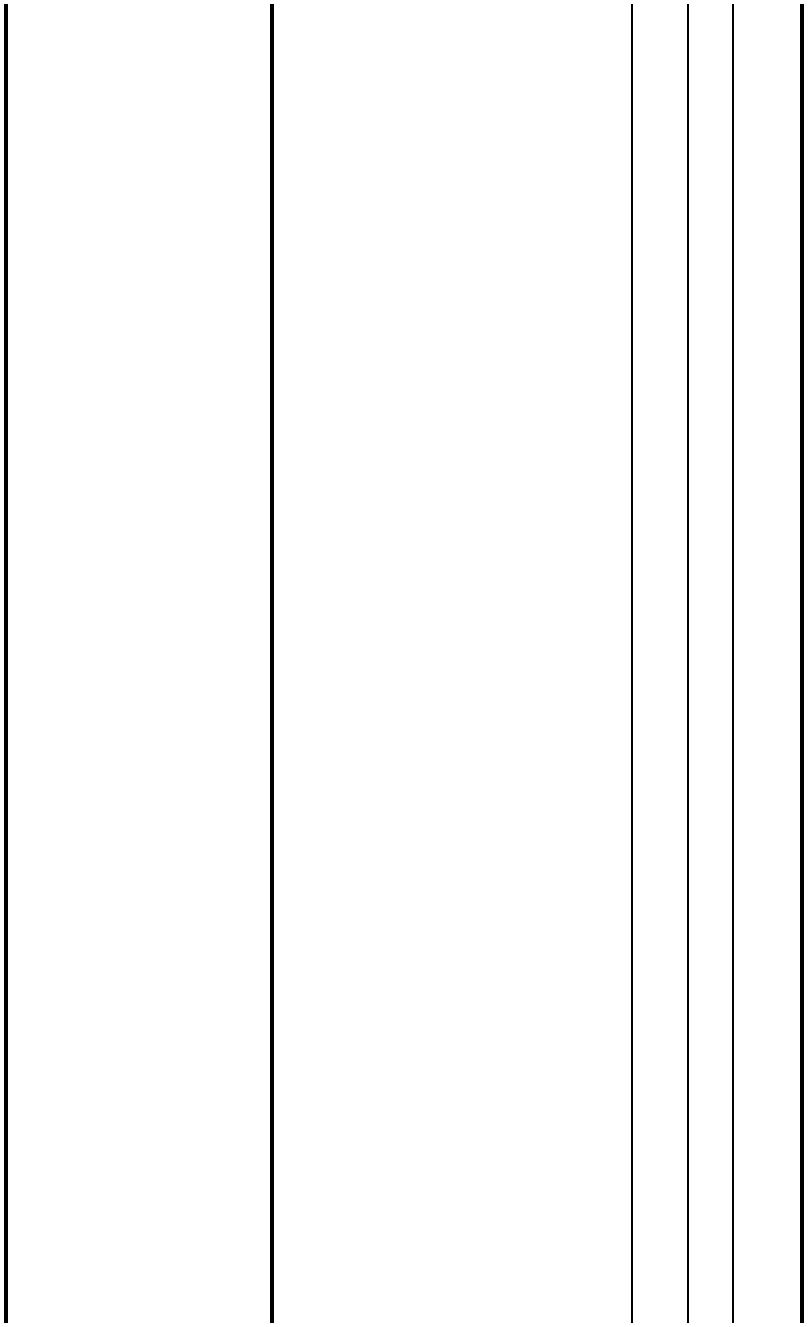


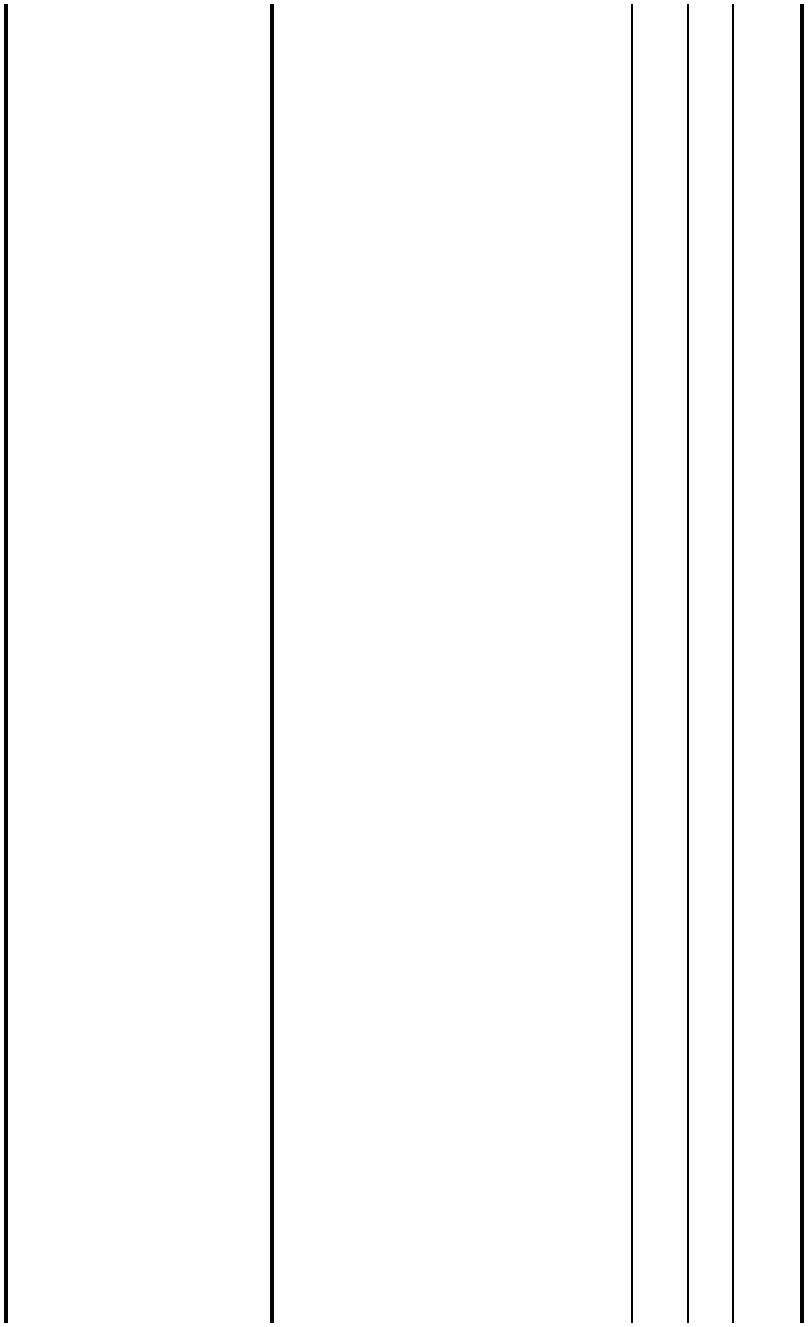


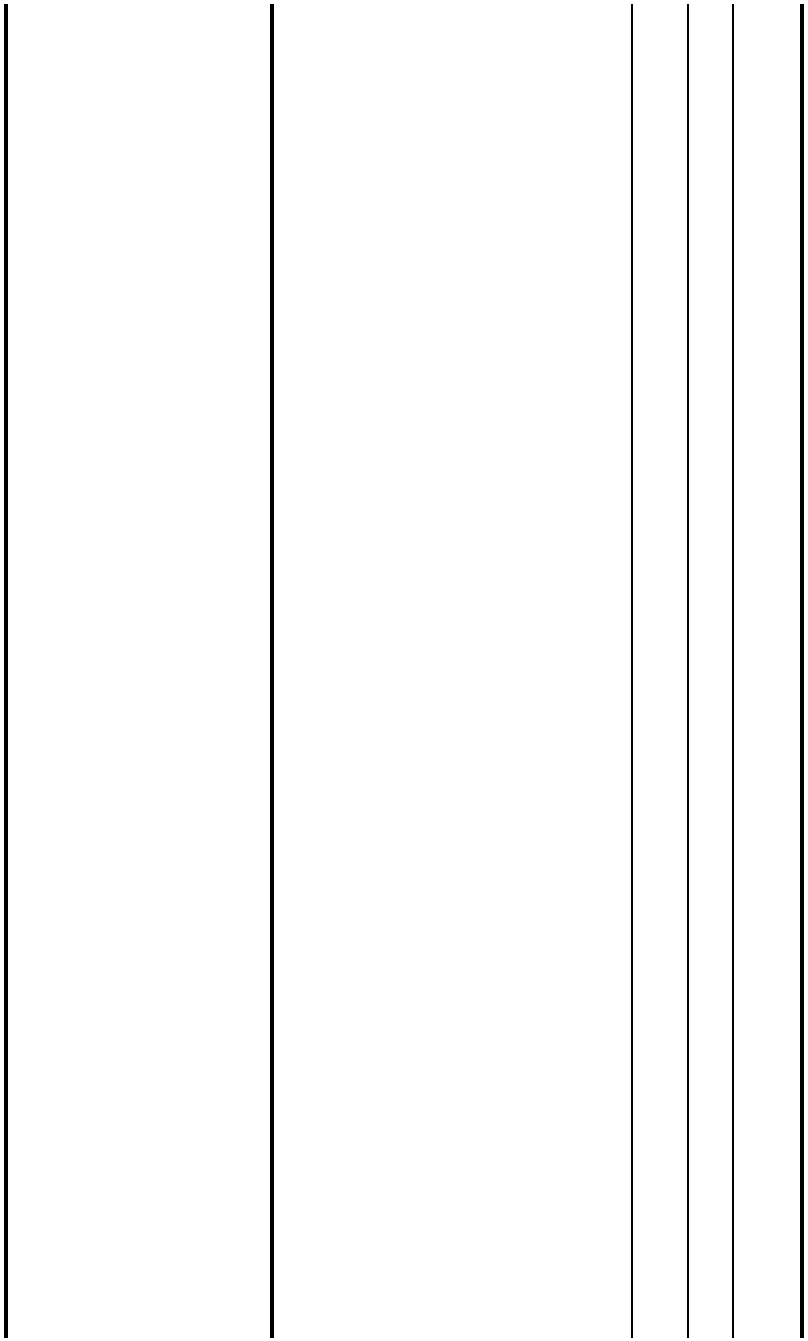


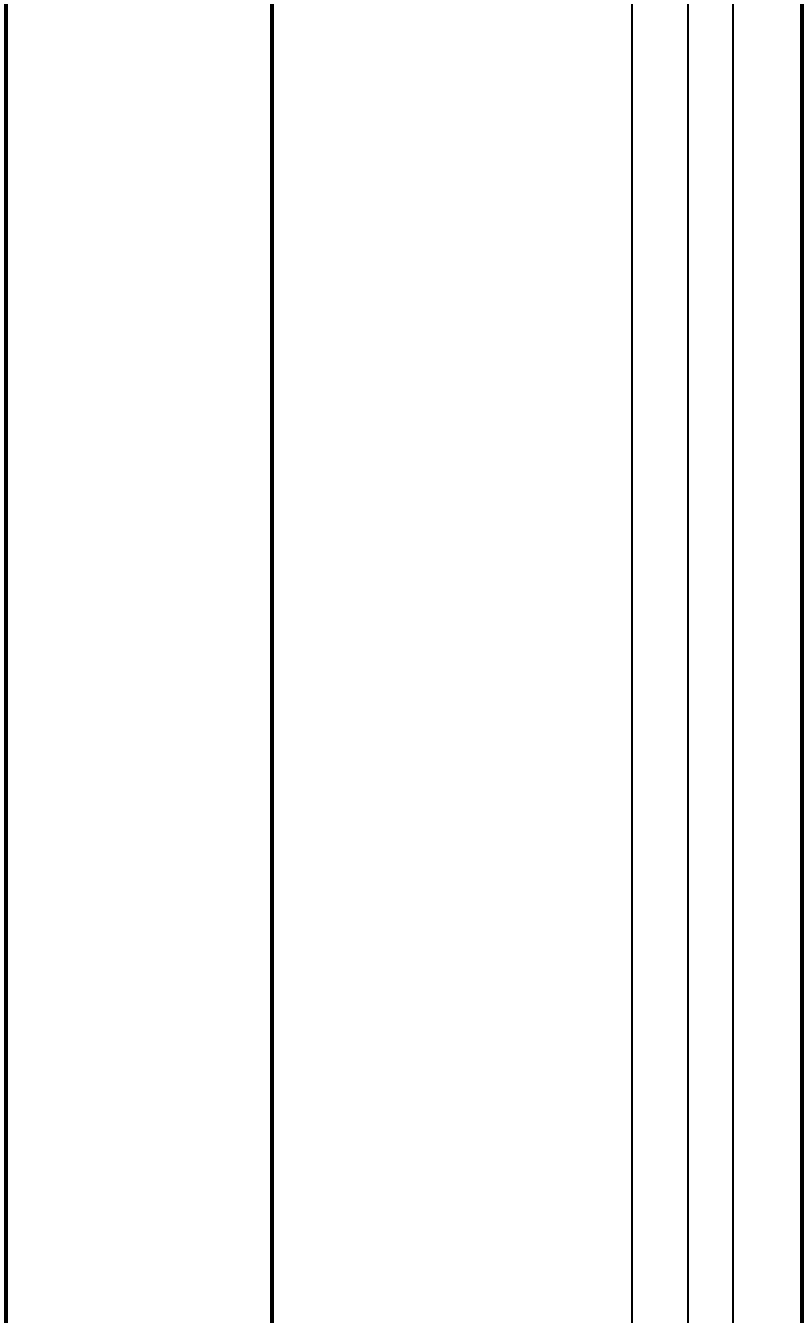


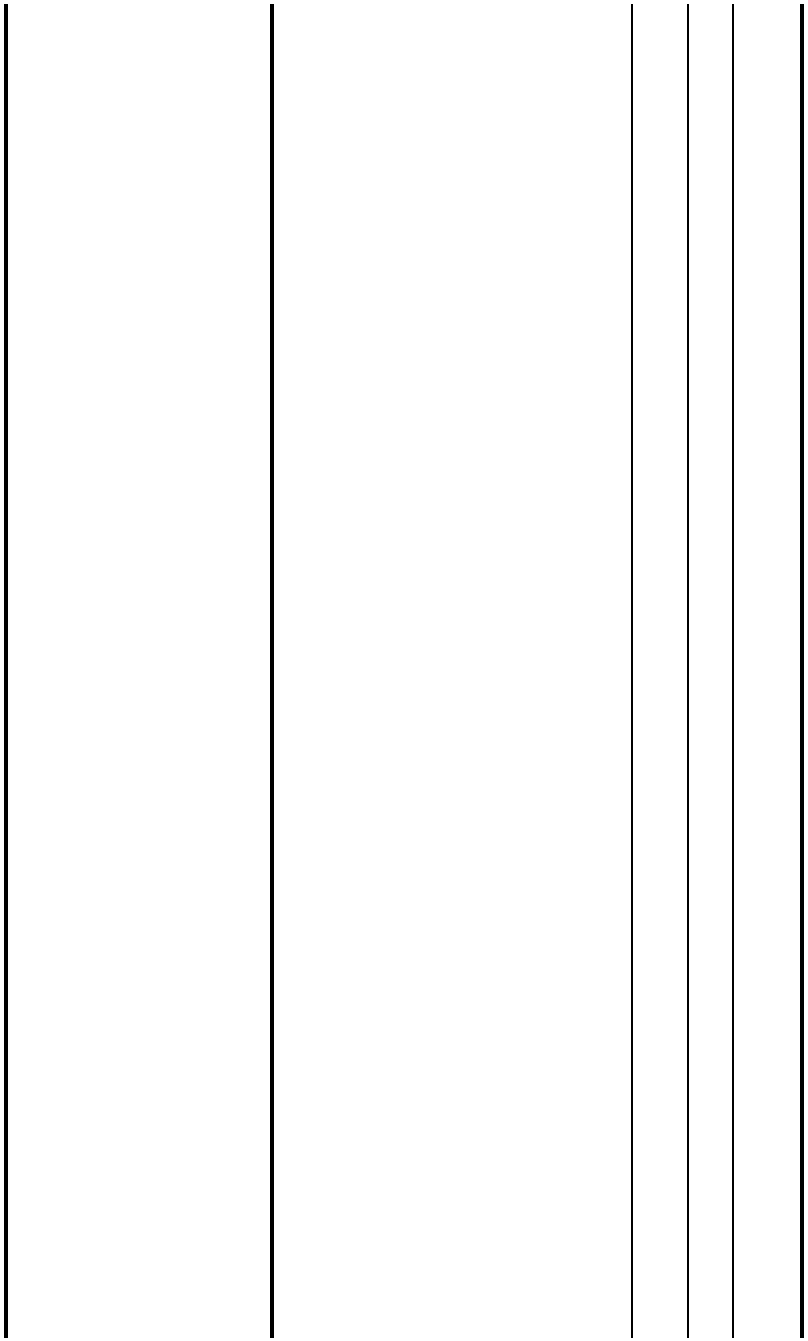


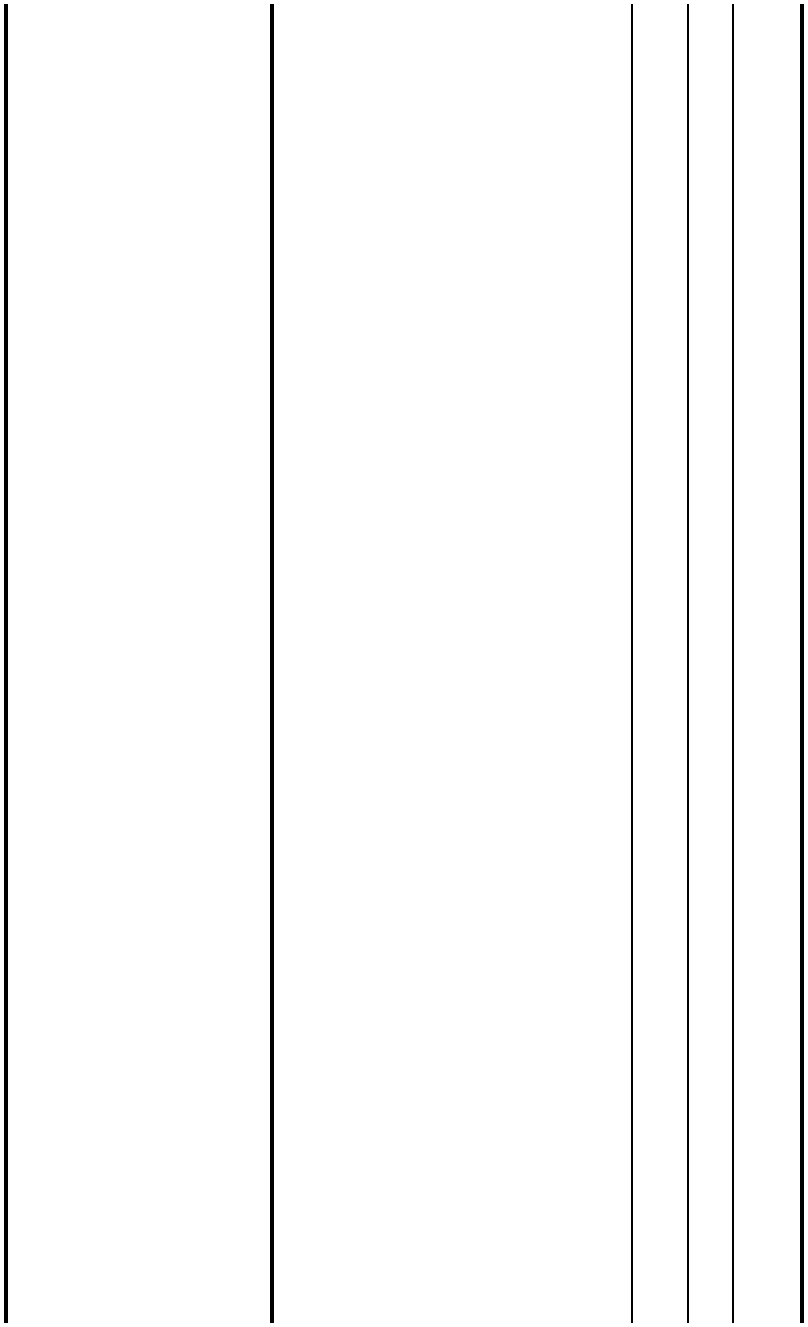


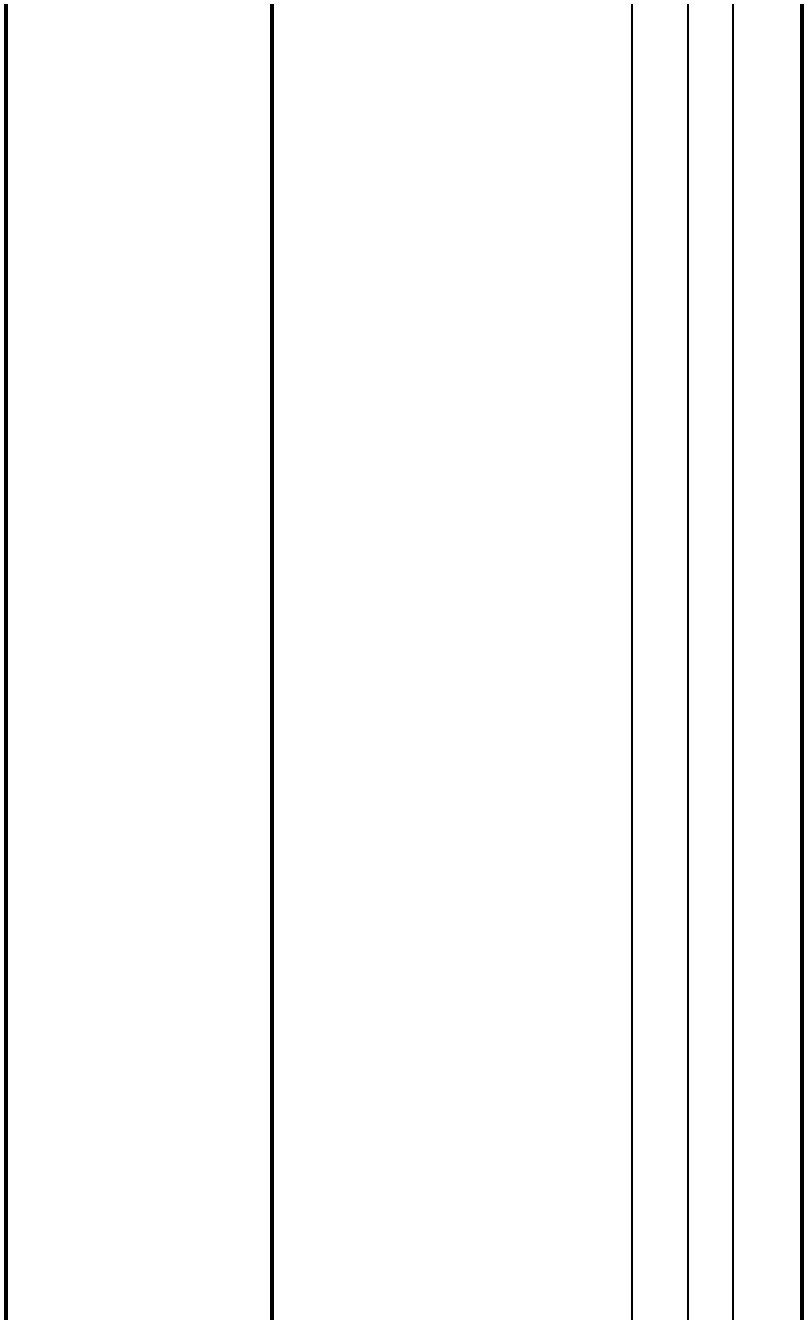


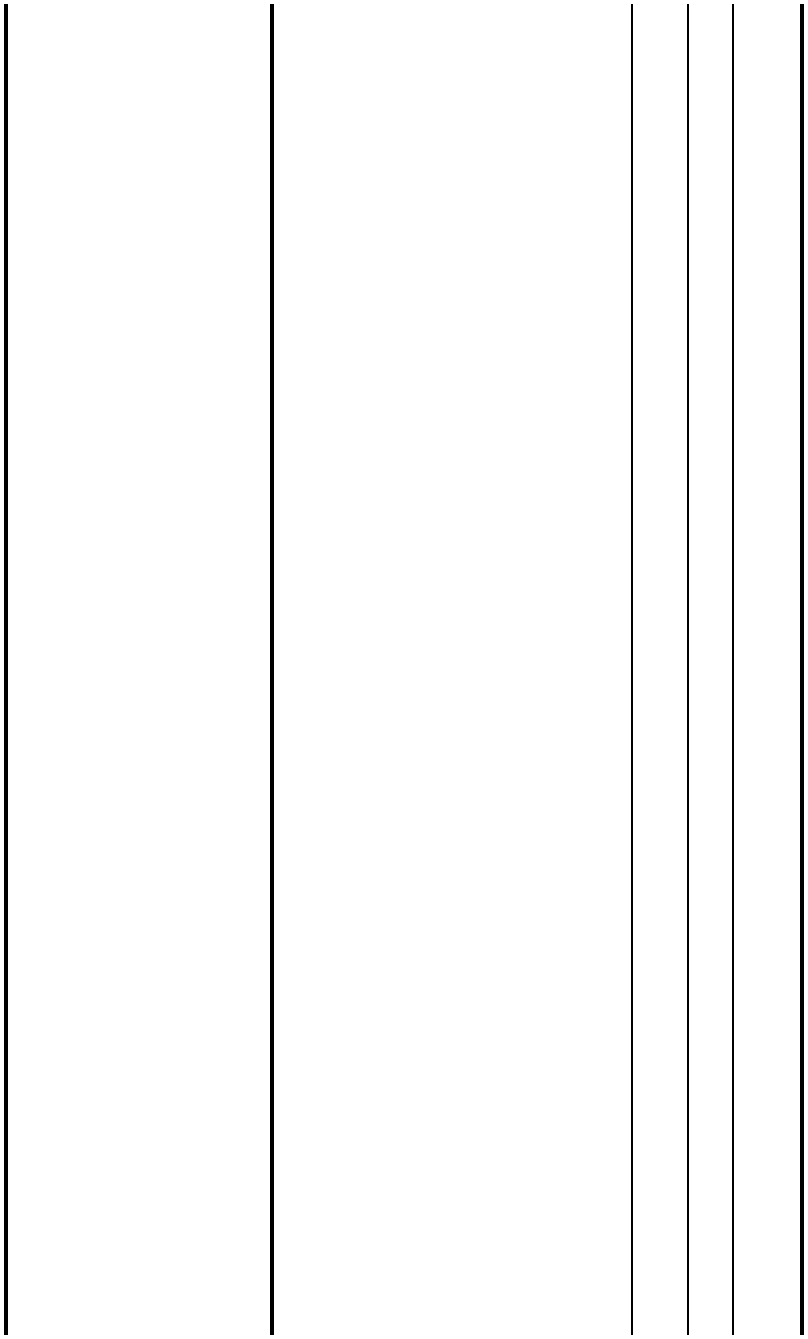


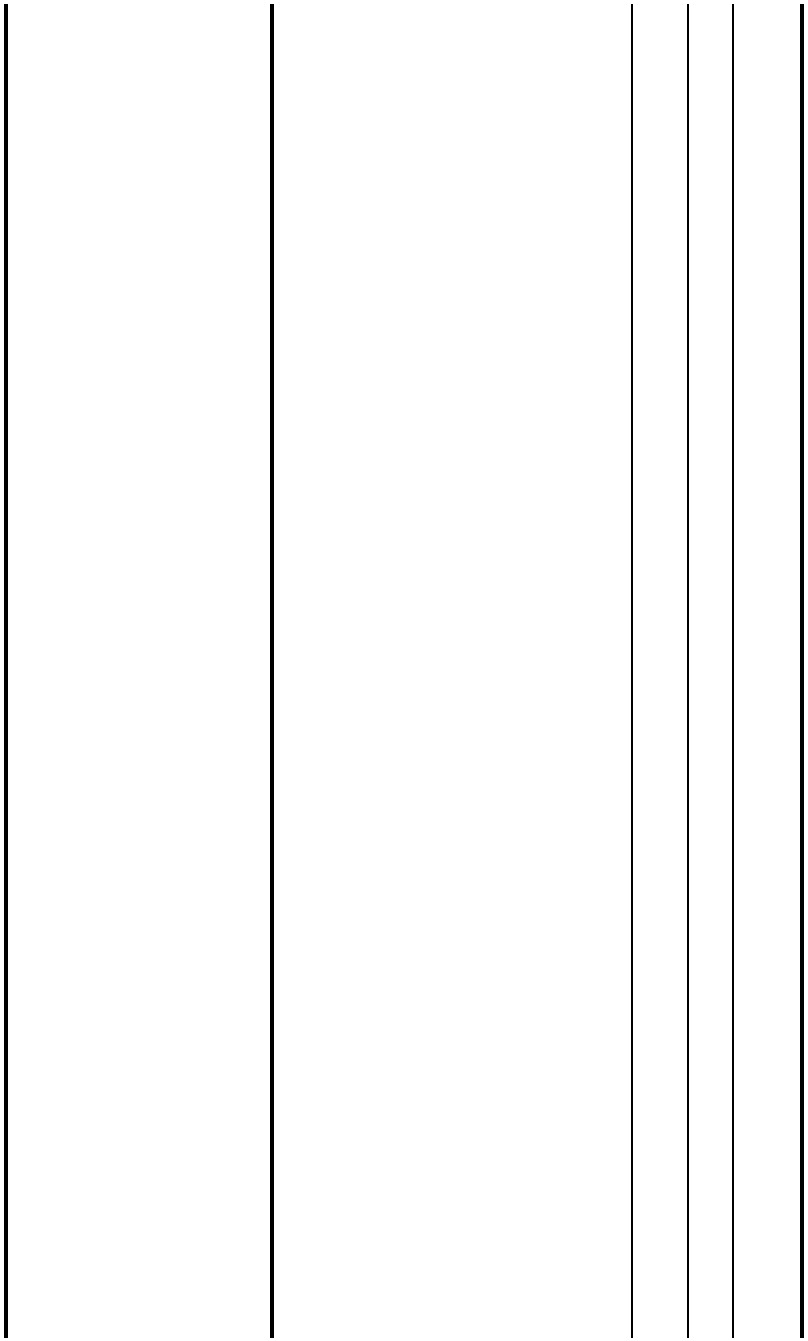


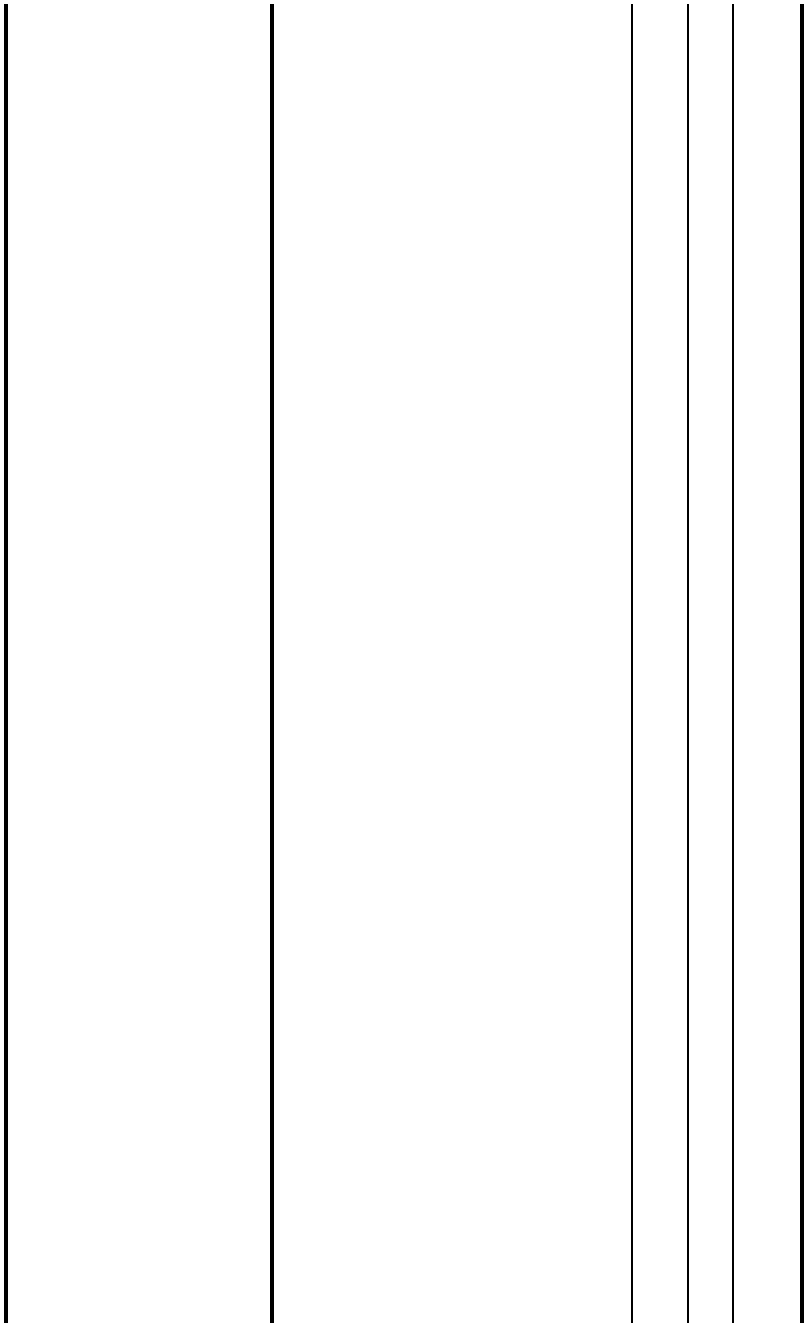


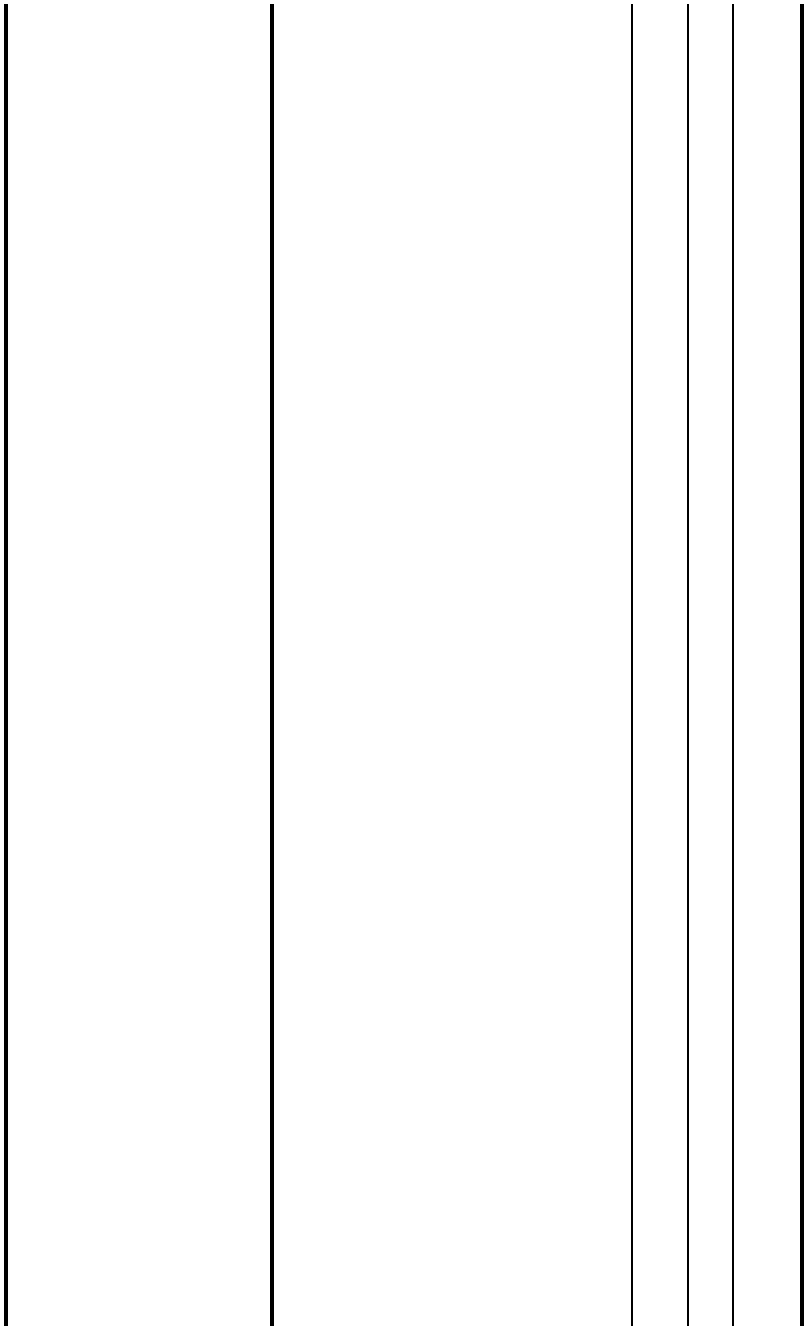


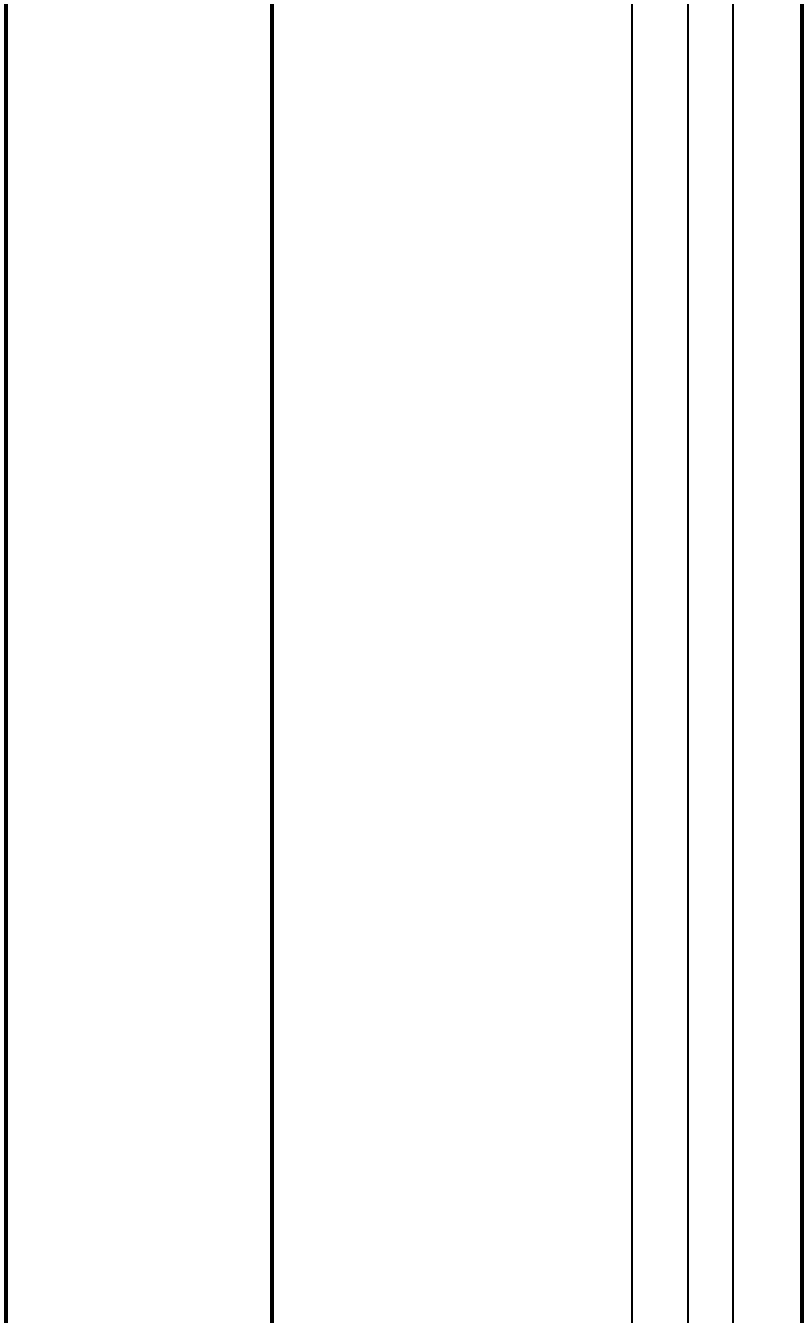


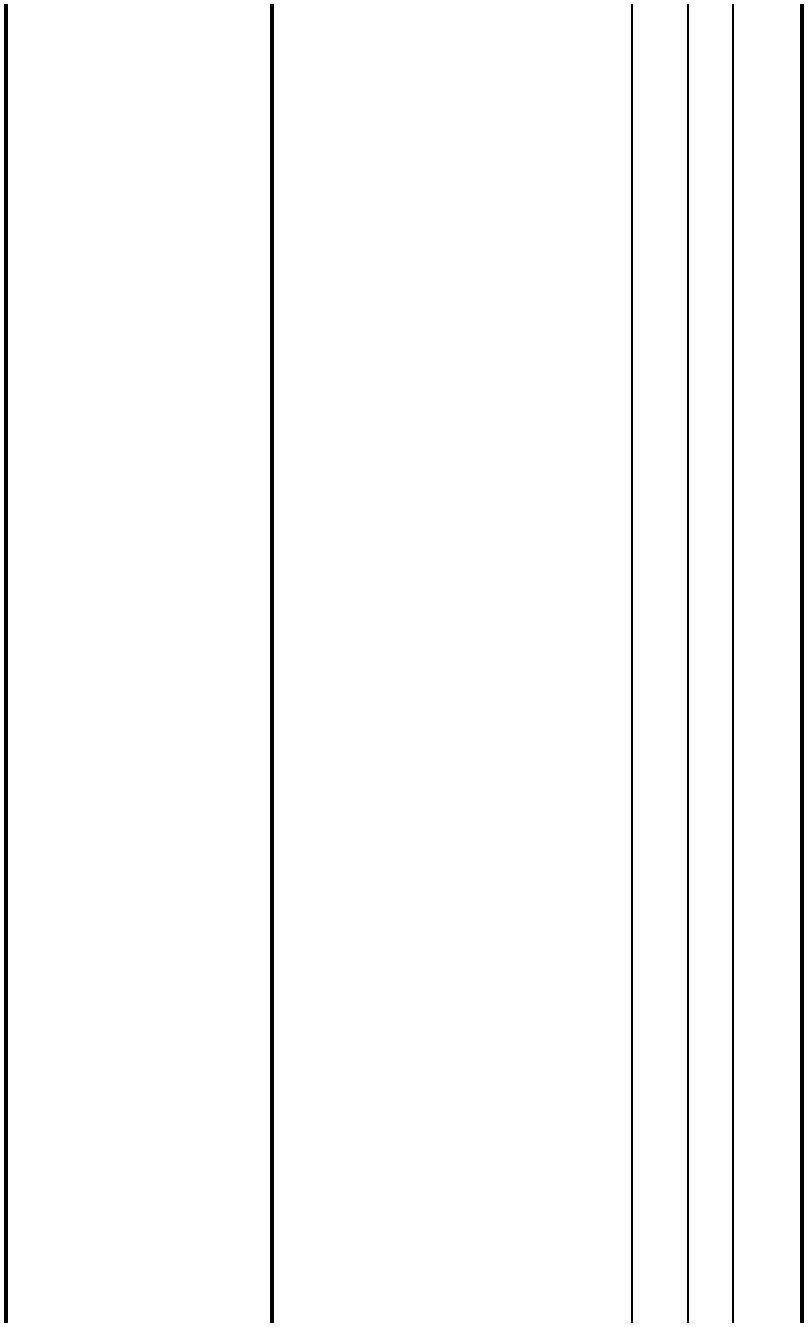


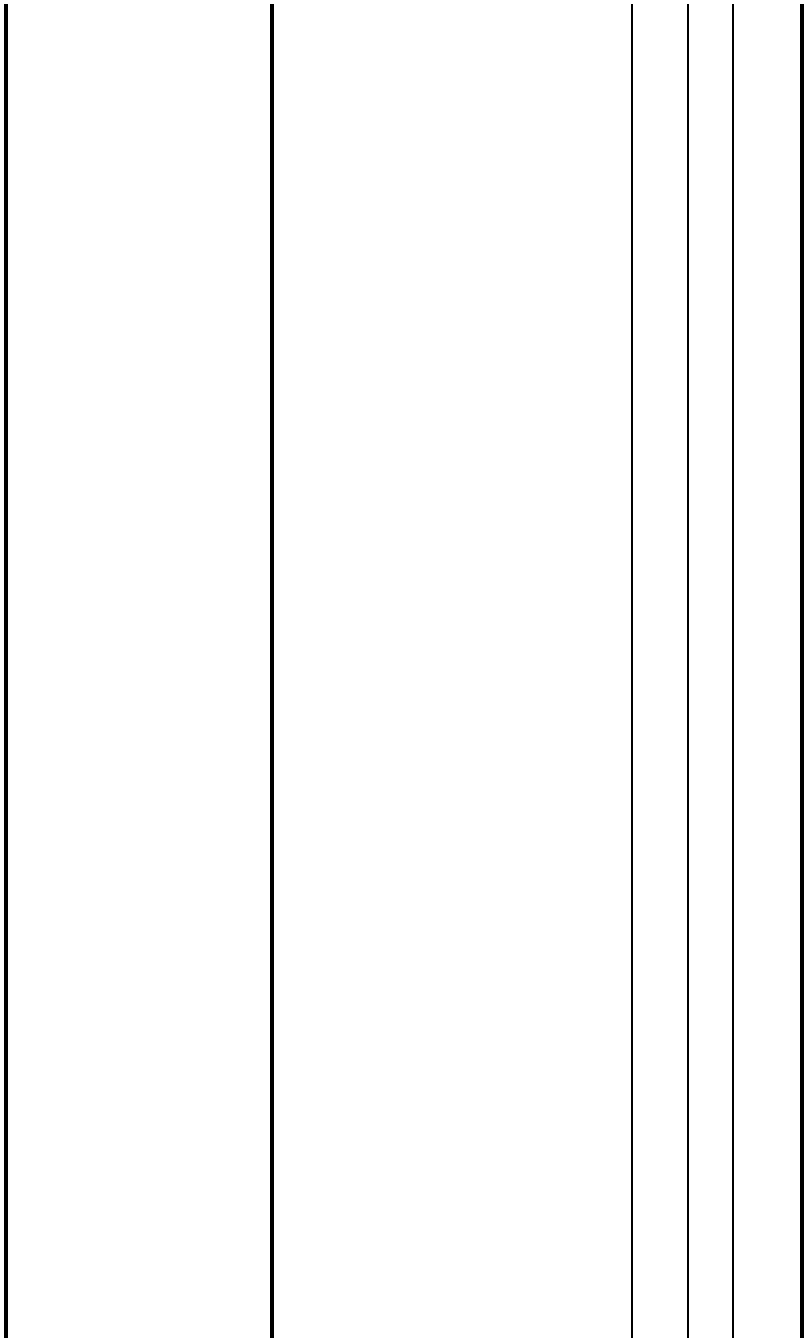


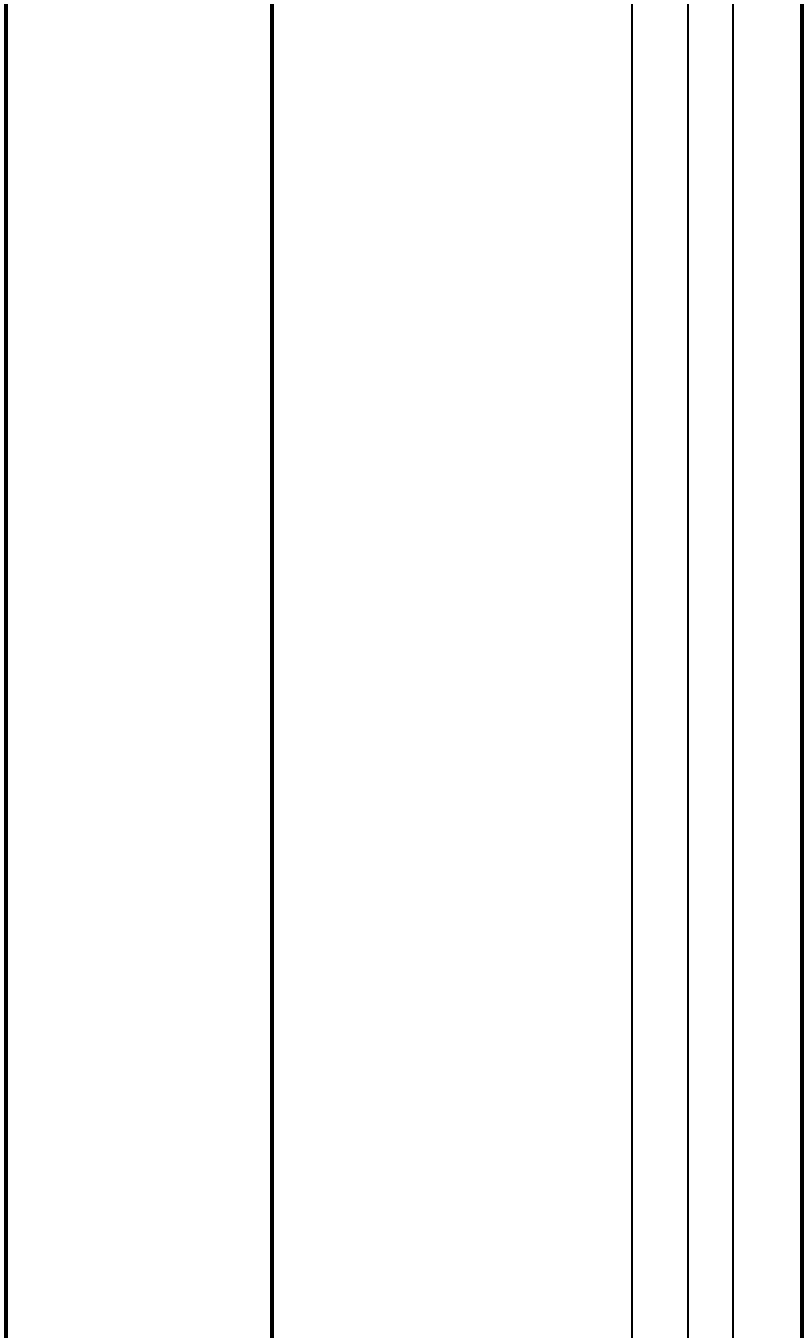


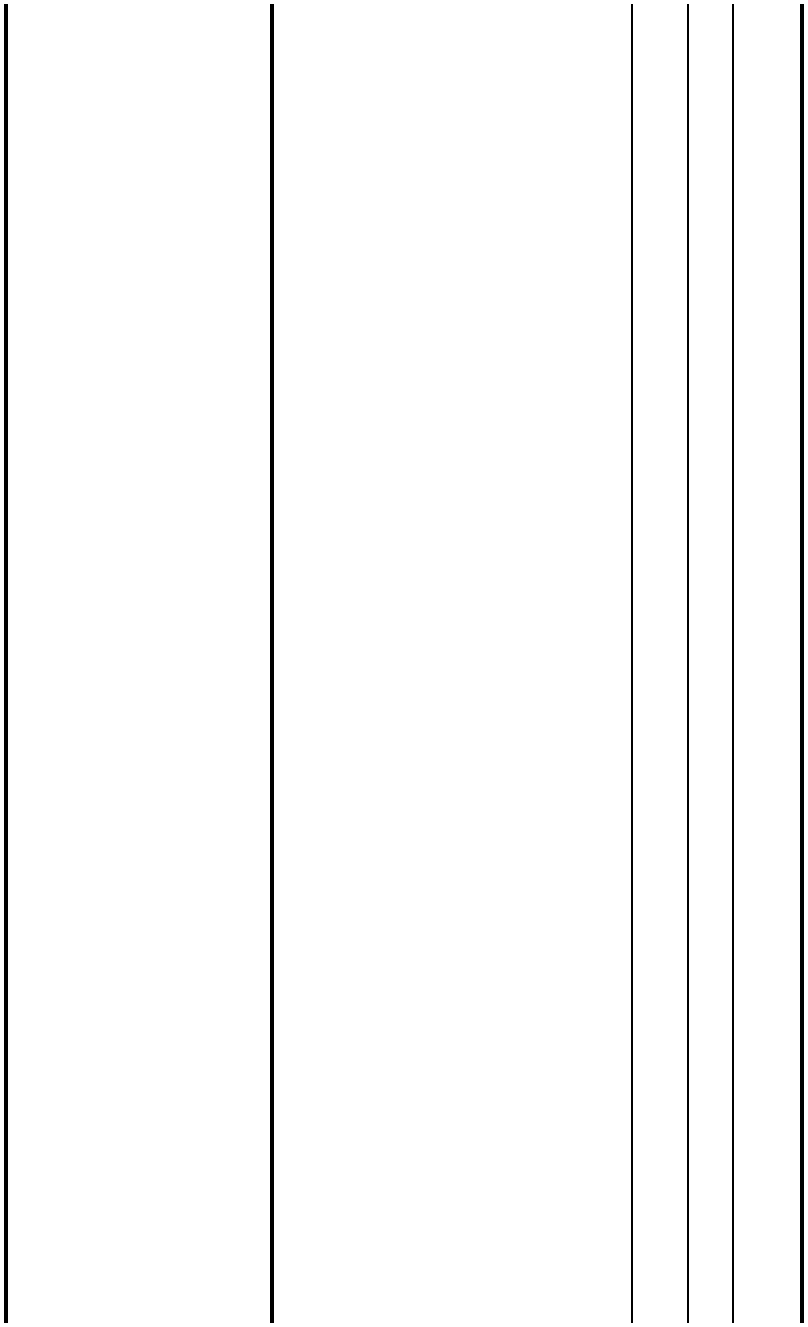


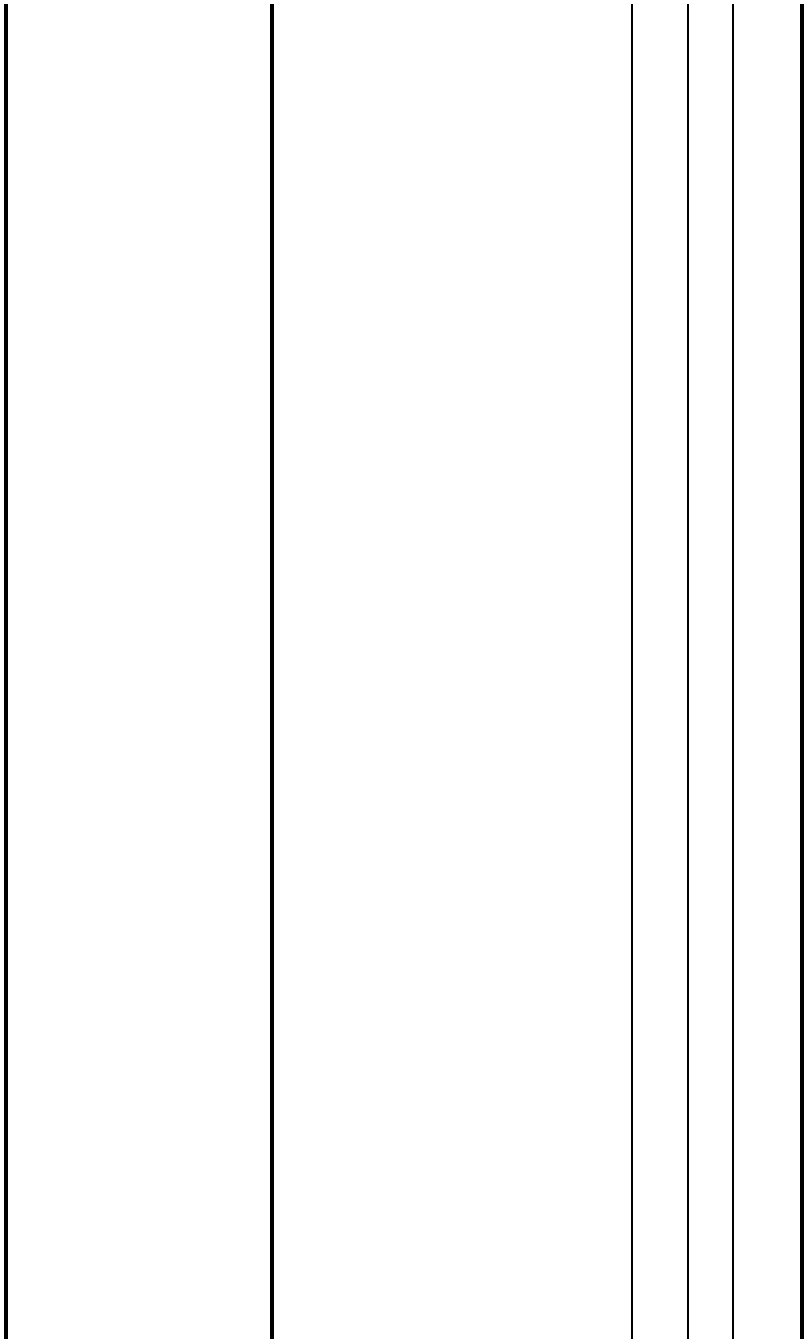


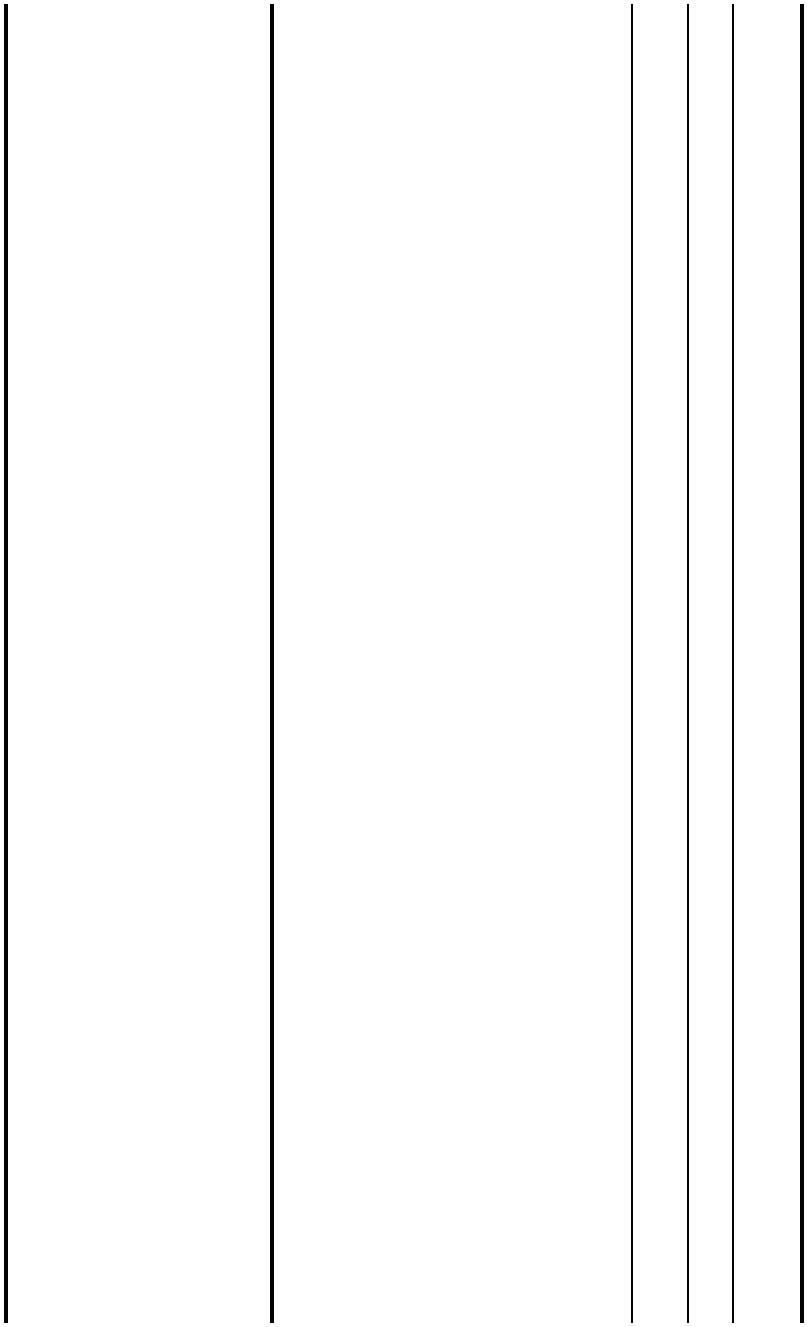


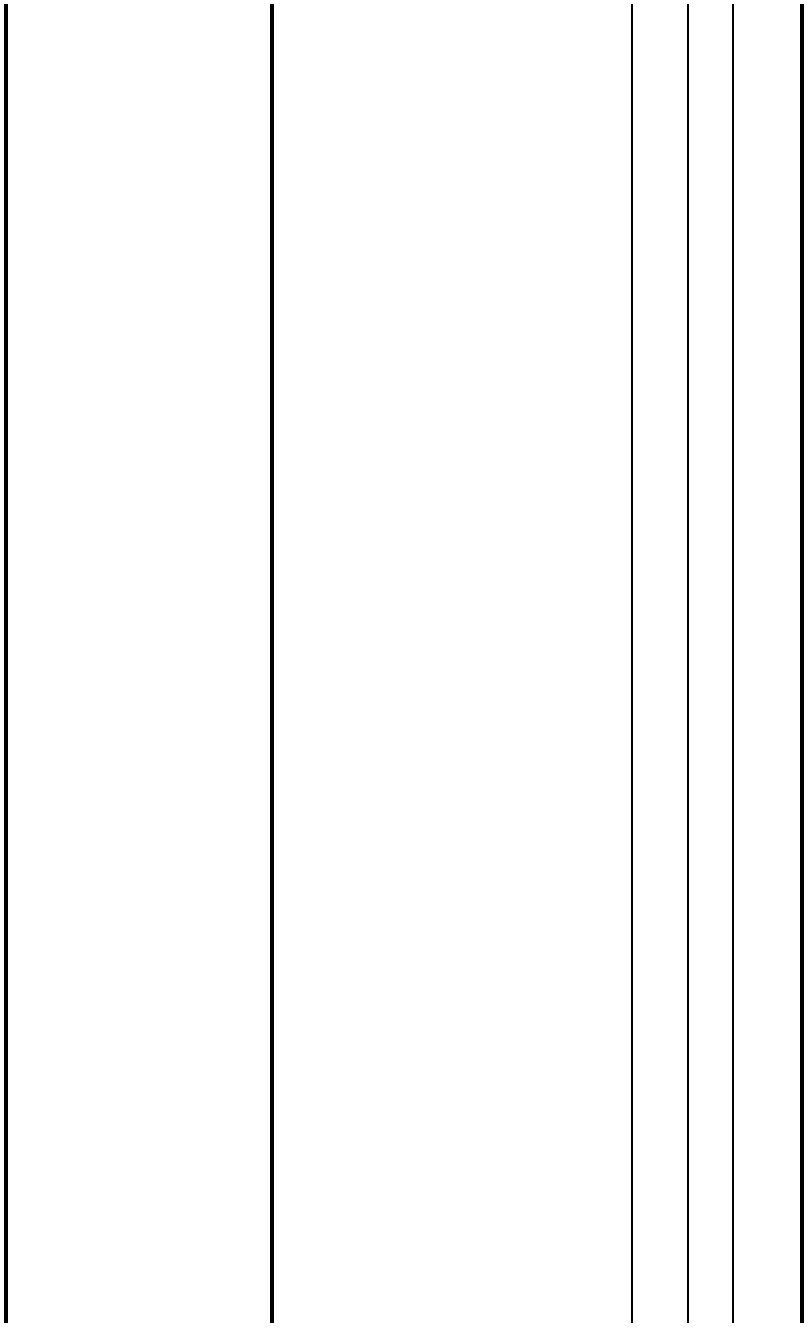


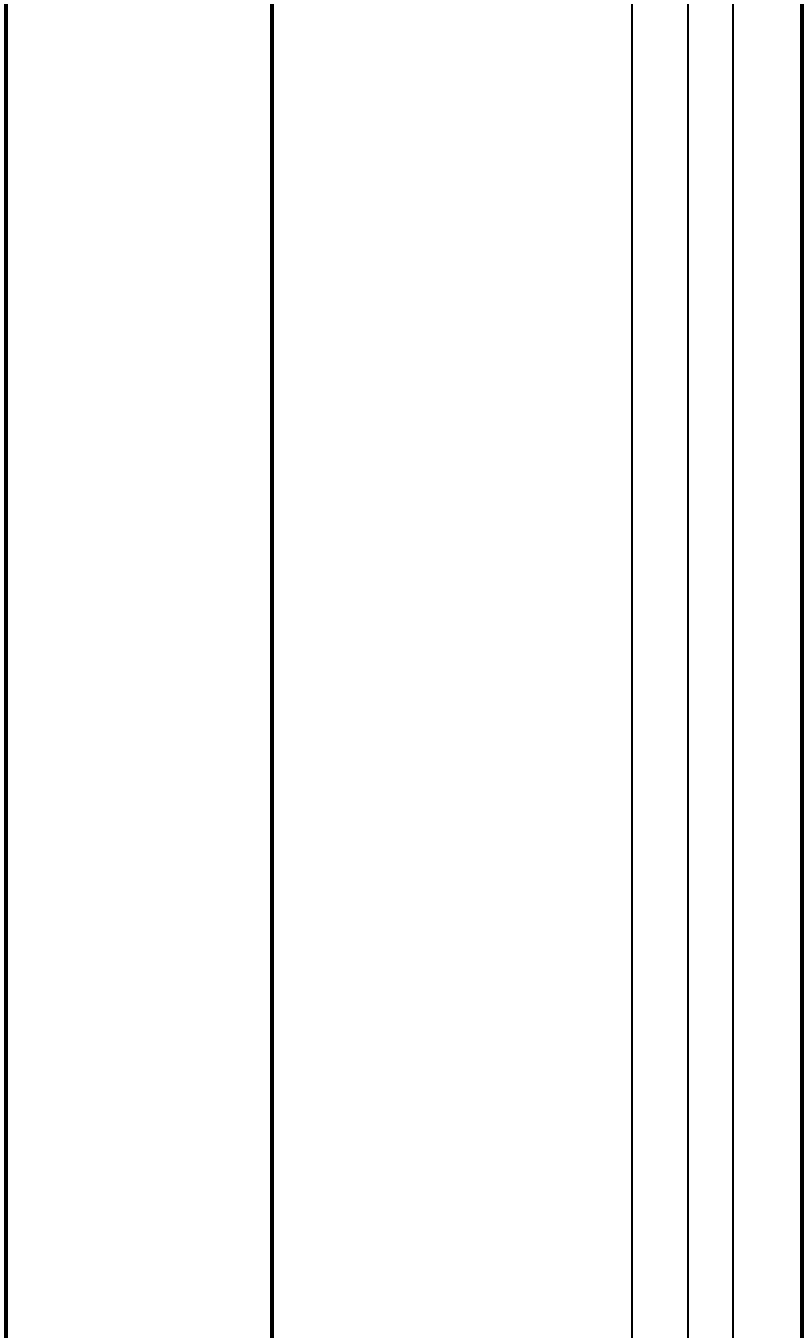


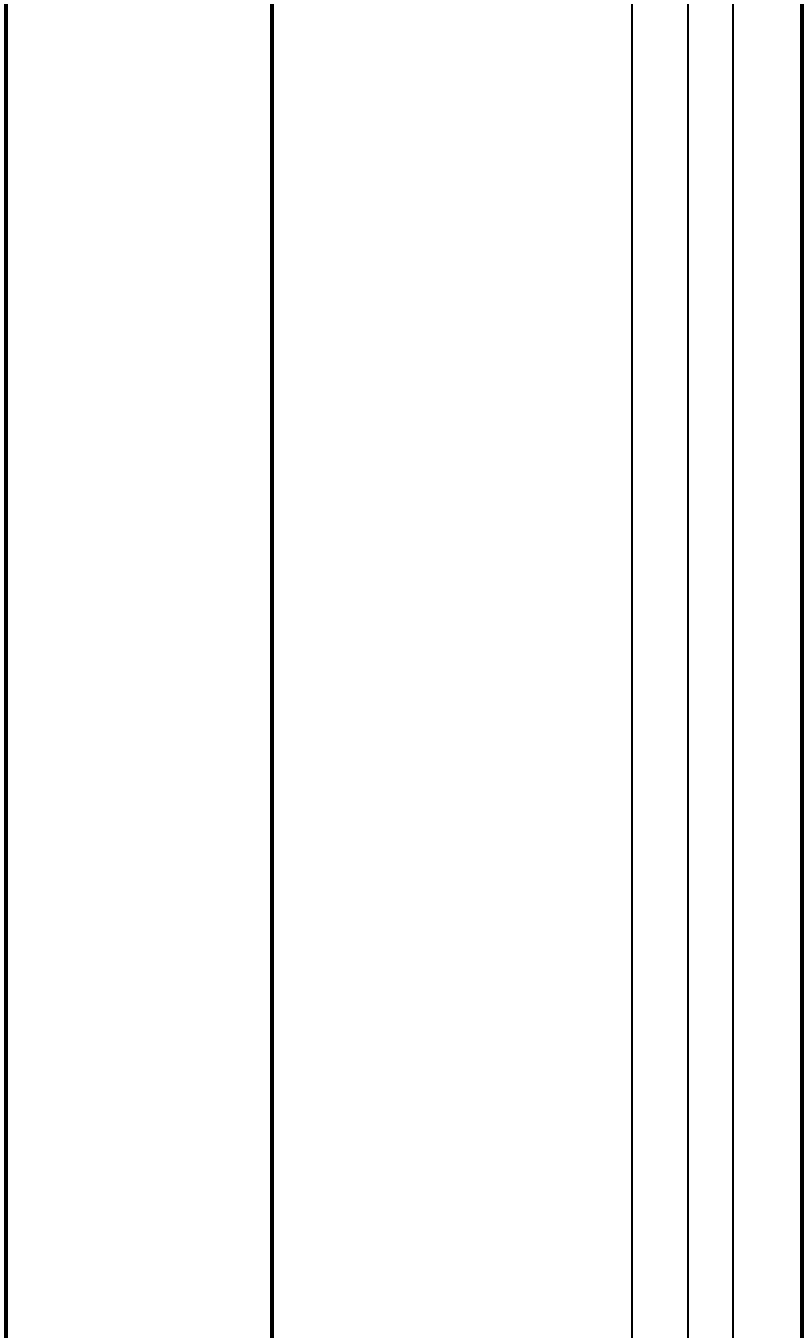


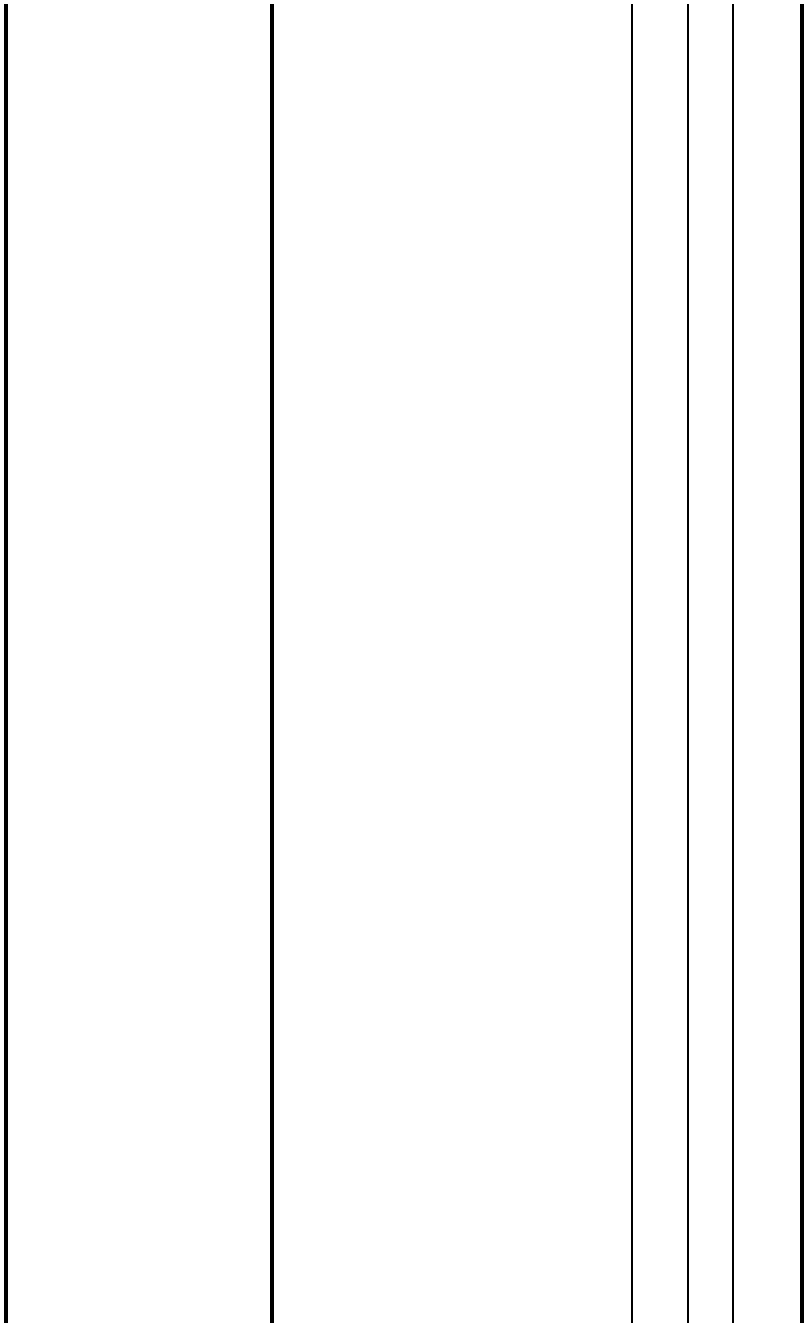


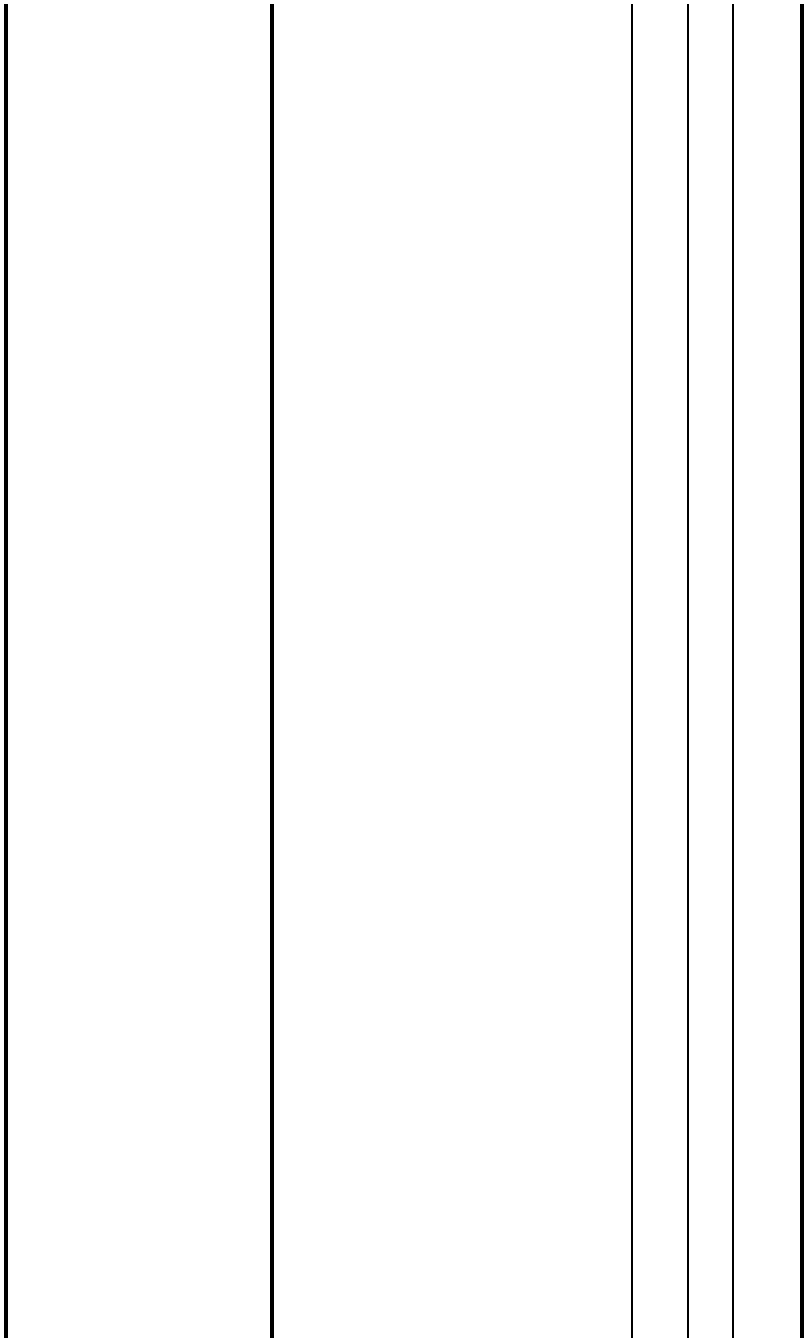


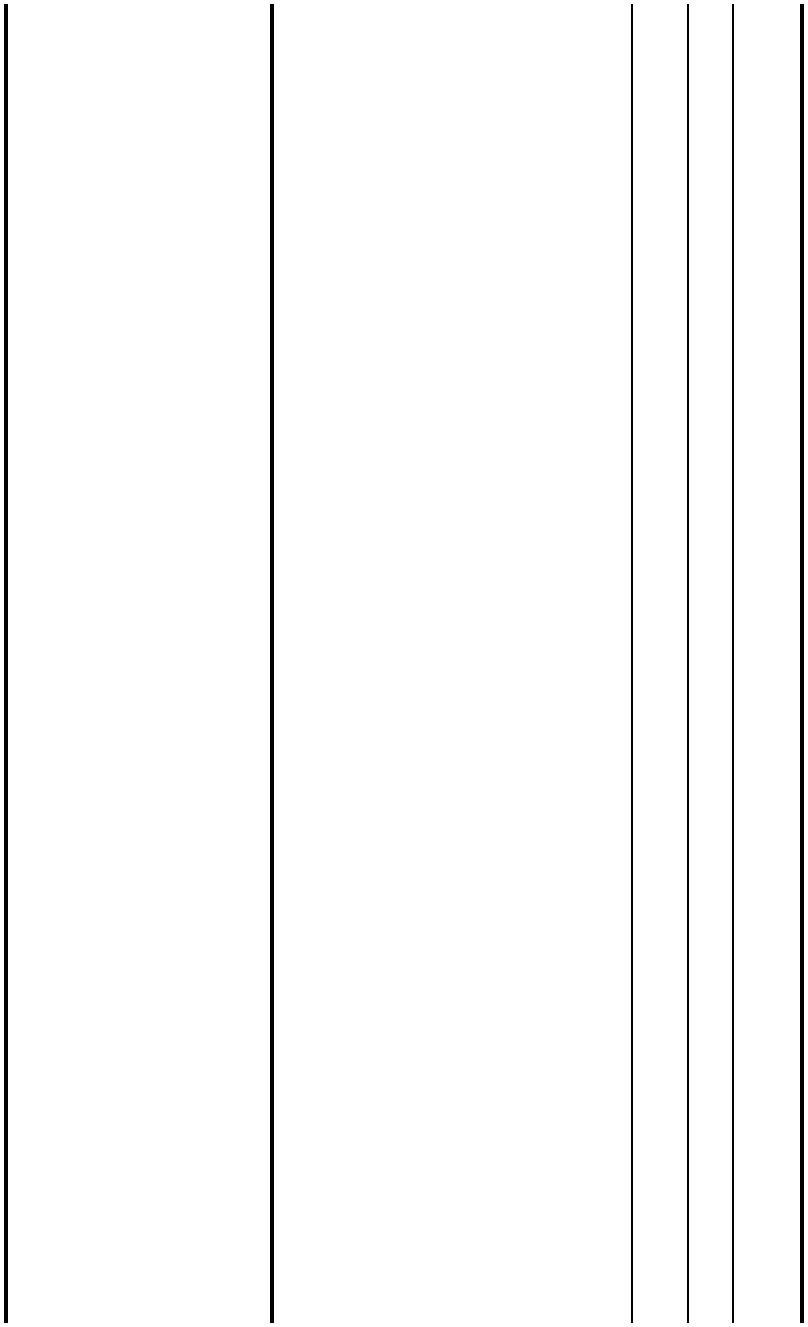


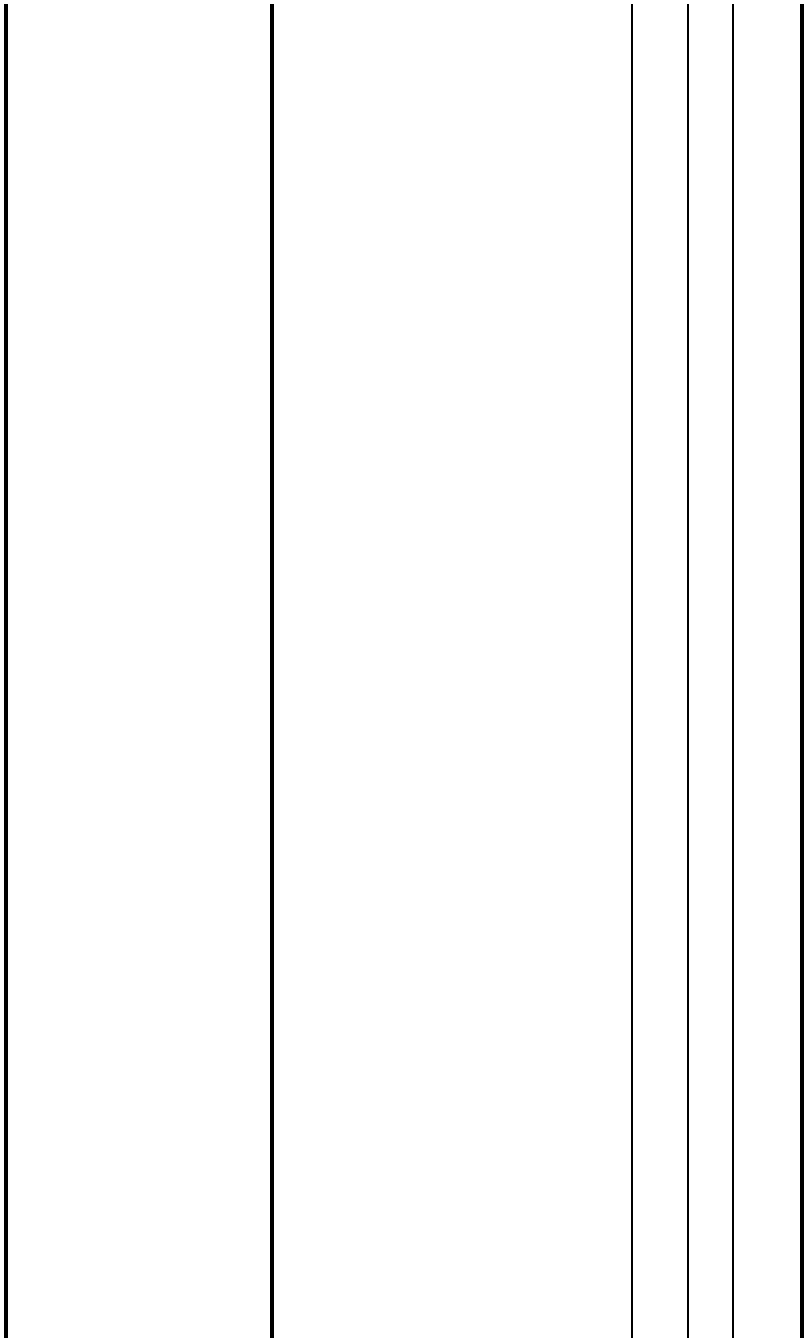


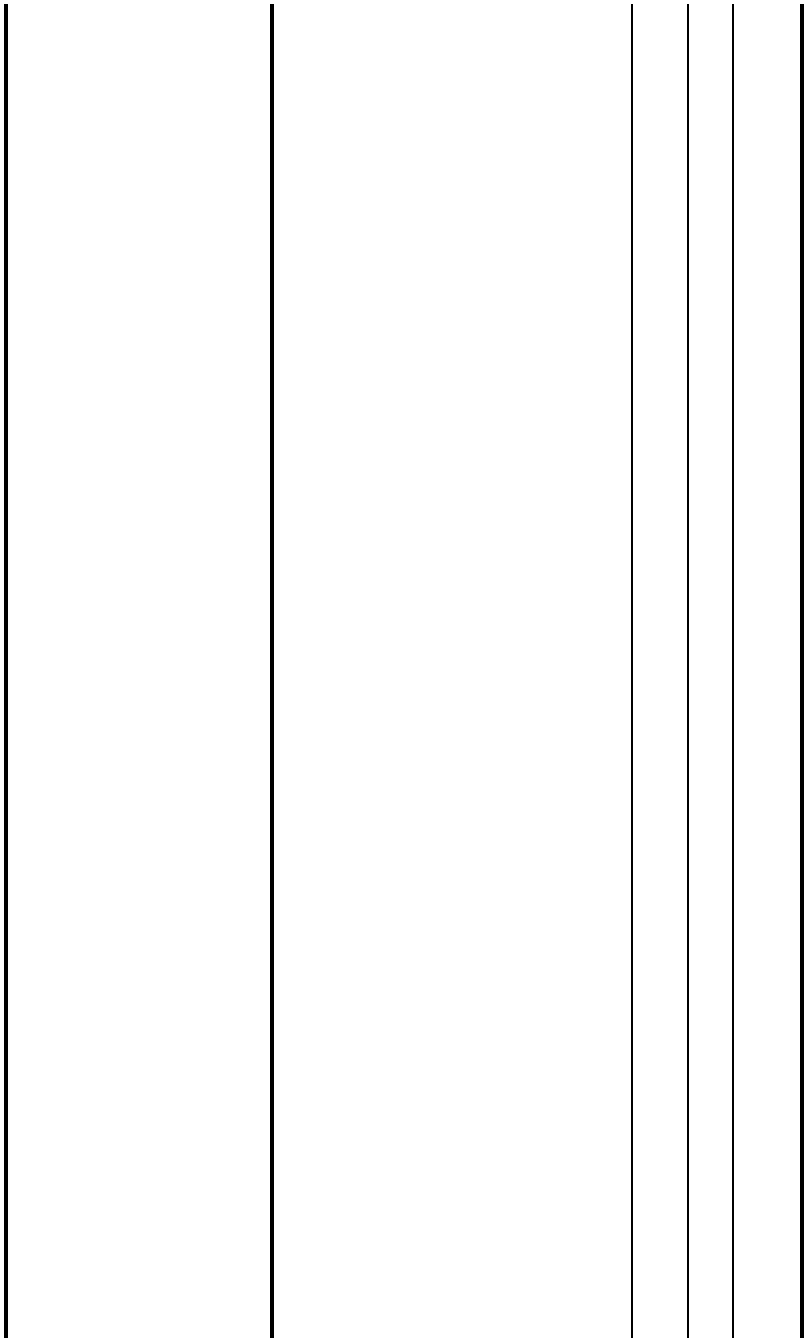


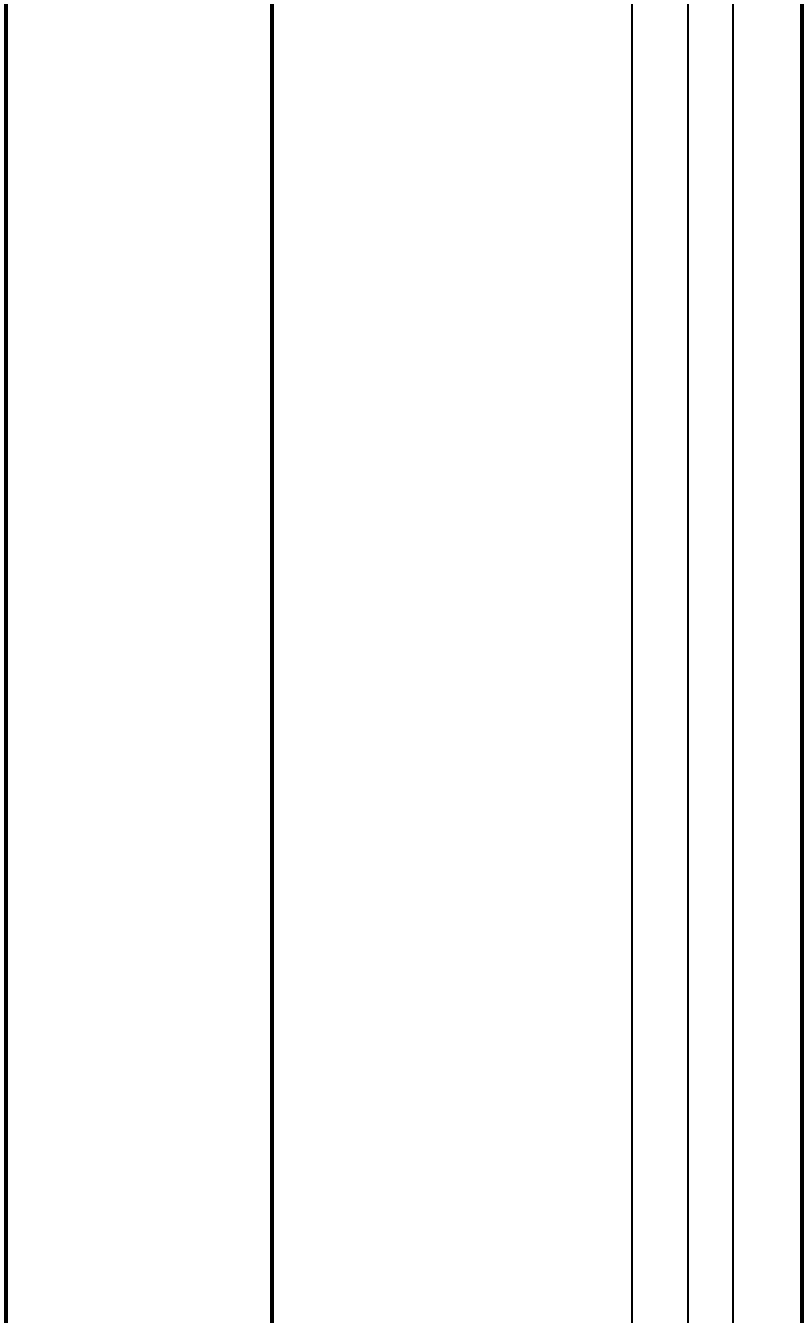


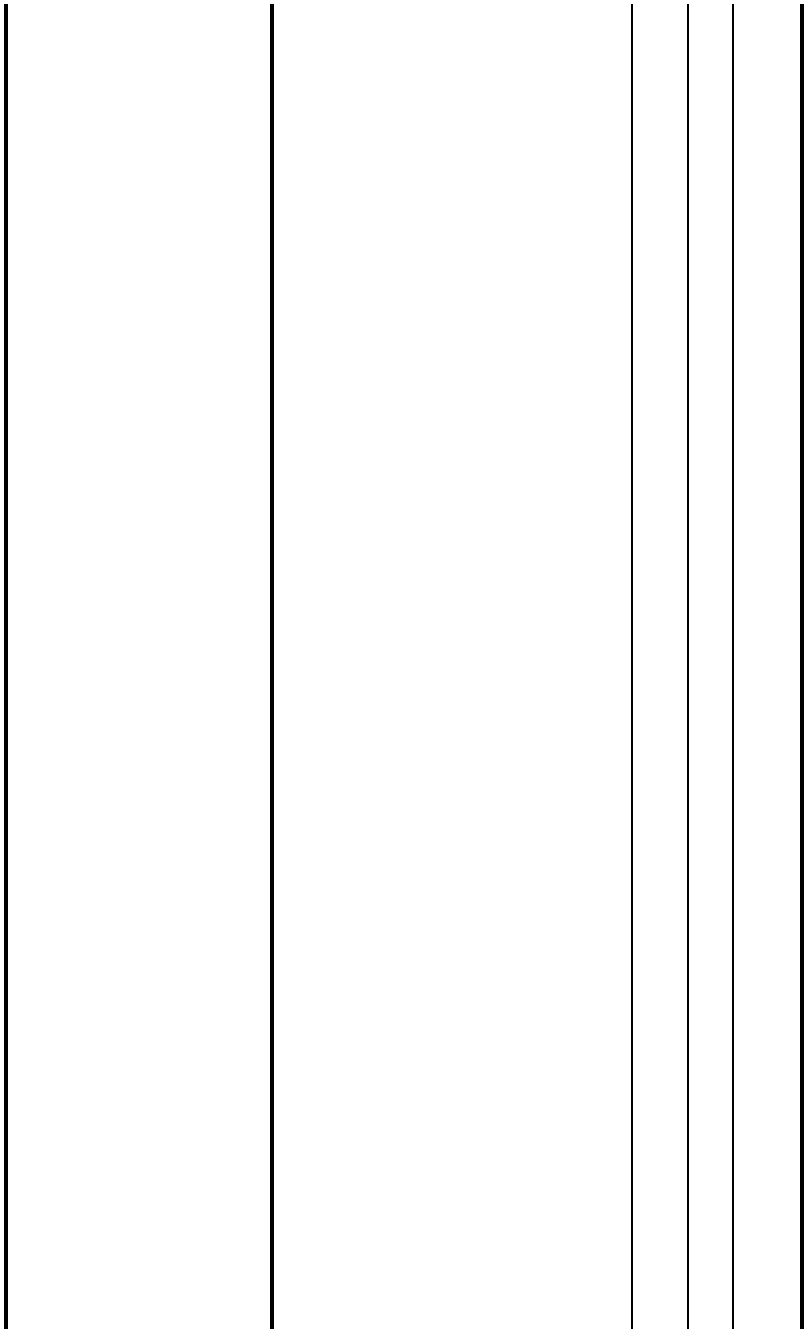












Notes:

[1] Binary values are read left-to-right, or msb first.

[2] Hexadecimal digits can be 0-9, and a-f.

[3] *M* is capitalized in abbreviation to differentiate from *m*, which denotes milli or 10e–3, as in mHz (milliHertz).

See Also

Related Documentation Topics

Documentation Assumption of Previous Knowledge

Documentation Organization

Documentation Conventions

Documentation References and Additional Sources

Relationship Between Online and Printed Documentation

The Signalogic DSP Software System Guide, Users Guide, Reference Guide, and Tutorial have

both an online Help form and a printed form. These forms are actually created from the same source, and so will appear similar. However, there are significant differences, as listed below:

Chapter and section numbers are not shown in online Help, either at the start of chapters and sections or when referring to other information. In online Help, references to other chapters and sections are instead made by highlighting the reference as "jumptext", which can be selected by the mouse.

Frequently, at the end of an online Help section, there appears a list of additional topics, which are shown as jumptext. These lists of "Related Topics" do not appear in the printed documentation.

Note that the native online Help inside Hypersignal-Macro series software is completely separate from other Hypersignal-Macro series software documentation. Hypersignal native online Help is intended for user mode operation only, and is most closely related to the Hypersignal-Macro Series

Software Reference Guide. It shows mainly a summary of valid menu field-entries at various stages of Hypersignal program operation. When a Hypersignal package is used as a DSP Engine with DSPower series software, the Hypersignal native online Help is not typically activated.

Related Documentation Topics

Documentation Assumption of Previous Knowledge

Documentation Organization

Documentation Conventions

Differences Between Online and Printed Documentation

Online Documentation Conventions

Documentation References and Additional Sources

Online Documentation Conventions

Jumps to other topics ("jumptext")are highlighted in green in online Help.

Glossary terms are highlighted in purple in online Help.

Often times, jumptext will have a small "secondary window" icon to its immediate right. Selecting this icon instead of the jumptext itself will cause the new topic to be displayed in a

secondary help window.

Sequential viewing of online Help topics can be done using the "browse" buttons in online Help display, or using the DSPower Navigator to view preceding and succeeding topics.

Frequently, at the end of an online Help section, there appears a list of additional topics, which are shown as jumptext. If a topic is bold, but not the jumptext color (green), then this signifies the current topic (topic you are already in).

Related Documentation Topics

Documentation Assumption of Previous Knowledge

Documentation Organization

Documentation Conventions

Differences Between Online and Printed Documentation

Online Documentation Conventions

Documentation References and Additional Sources

2.4 Documentation References and Additional Sources

References and Additional Sources Of Documentation

Related Documentation Topics

Documentation Assumption of Previous Knowledge

Documentation Organization

Documentation Conventions

Relationship Between Online and Printed Documentation

Online Documentation Conventions

Documentation References and Additional Sources

DSP Theory and Practical Application

A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing* (Englewood Cliffs, NJ: Prentice Hall), 1975.

E.O. Bingham, *The Fast Fourier Transform and its Applications* (Englewood Cliffs, NJ: Prentice Hall), 1988.

A. Peled and B. Liu, *Digital Signal Processing* (New York, NY: John Wiley & Sons, Inc.), 1976.

L.B. Jackson, *Digital Filters and Signal Processing* (Boston, MA:

Kluwer Academic Publishers), 1986.

Hypersignal-Macro Users Manual, "Brief Discussion of Sampled Data Systems" (Dallas, TX: Signalogic) March, 1993.

References and Additional Sources Of Documentation

DSP Theory and Practical Application

DSP/Analog and Acquisition Hardware

Waveform File Formats

Software Packages

DSP Chip Programming

DSP/Analog and Acquisition Hardware

Hypersignal-Macro Hardware Reference Guide (Dallas, TX: Signalogic), December, 1994.

References and Additional Sources Of Documentation

DSP Theory and Practical Application

DSP/Analog and Acquisition Hardware

Waveform File Formats

Software Packages

DSP Chip Programming

Waveform File Formats

Hypersignal-Macro Users Manual, "Hypersignal Waveform File

Format" (Dallas, TX: Signalogic), March, 1993, chapter 5.

Microsoft Windows Multimedia Programmer's Reference (Redmond, WA: Microsoft Press), 1991, chapter 8.

References and Additional Sources Of Documentation

DSP Theory and Practical Application

DSP/Analog and Acquisition Hardware

Waveform File Formats

Software Packages

DSP Chip Programming

Software Packages

MATLAB User's Guide (Natick, MA: The MathWorks), 1993.

Hypersignal-Macro Users Manual, (Dallas, TX: Signalogic) March, 1993.

References and Additional Sources Of Documentation

DSP Theory and Practical Application

DSP/Analog and Acquisition Hardware

Waveform File Formats

Software Packages

DSP Chip Programming

DSP Chip Programming

WE DSP32C Application Software Library Reference Manual (Allentown, PA: AT&T), 1989.

P. Papamichalis ed., *Digital Signal Processing Applications with the TMS320 Family* (Dallas, TX: Texas Instruments), 1990, vol. 3.

W.H. Press, W.T. Vetterling, S.A. Teukolsky, B.P. Flannery ed., *Numerical Recipes in C*, 2nd ed. (Cambridge, UK: Cambridge University Press), 1992.

Signalogic Application Note #4, "Assembly Language Programming Comparison for TMS320C30, DSP32C, and DSP96002", (Dallas, TX: Signalogic), 1992.

References Additional Sources Of Documentation

DSP Theory and Practical Application

DSP/Analog and Acquisition Hardware

Waveform File Formats

Software Packages

DSP Chip Programming

3 Installation

Overview

This section covers installation of all Signalogic DSP Software packages using the Signalogic Software Setup Utility, which is a Windows-based program. Although Hypersignal-Macro series packages can be installed without the Software Setup Utility (see section 2.1, Installing Hypersignal, in the Hypersignal-Macro Series Users Manual, and/or section 3.2.5 below, Installing and Configuring Hypersignal Software For User Mode Operation), the same information is covered here.

NOTE: Be sure to look through section 3.1, Before You Install, before installing.

Main Topics

Before You Install

Signalogic DSP Software License Agreement

Media and Disk Space Requirements

Documentation and Media You Should Have Received

Installing Hypersignal-Macro Series Software

Installing DSPower Series Software

Configuration and Checkout of DSPower-HWLib Software

Configuration and Checkout of DSPower-Block Diagram Software

Installing and Configuring Hypersignal Software For User Mode Operation

Release Notes

Directory Structure Chart

Uninstalling

3.1 Before You Install

Before you install, please read the Signalogic DSP Software License Agreement, if you have not already done so. Also, please check the requirements listed in section 1.3, System Requirements.

NOTE: Make sure you fill out and send in your product registration card. Registration ensures you will be notified of product updates and have the fastest possible technical support response.

You may also want to make a backup copy of the diskettes you have received. The Signalogic DSP Software License Agreement allows you to copies to be made for backup and archival purposes only. For more information, see section 3.1.1, Signalogic DSP Software License Agreement.

In addition to the license agreement, you may want to read the sections below before you install your software:

1. section 3.1.2, Media and Disk Space Requirements
2. section 3.1.3, Documentation and Media You Should Have Received

3.1.1 Signalogic DSP Software License Agreement

Reprint of Signalogic DSP Software License Agreement follows. *Note that this agreement supersedes the License Agreement and limited warranty found in March, 1993 and earlier copies of the Hypersignal-Macro Users Manual.*

READ THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT CAREFULLY BEFORE OPENING THIS PACKAGE. THIS LICENSE AGREEMENT REPRESENTS THE ENTIRE AGREEMENT BETWEEN YOU (the "Licensee"--either an individual or entity) AND SIGNALOGIC, INC. ("SIG") CONCERNING THE COMPUTER SOFTWARE CONTAINED HEREIN ("PROGRAM"), AND THE ACCOMPANYING USER DOCUMENTATION.

BY OPENING THIS PACKAGE, YOU ACCEPT THE TERMS OF

THIS AGREEMENT. IF YOU ARE NOT WILLING TO DO SO, RETURN THE UNOPENED PACKAGE IMMEDIATELY FOR A FULL REFUND. YOU MAY ALSO RECEIVE A FULL REFUND IF YOU TERMINATE WITHIN THE TRIAL PERIOD OF THIRTY (30) DAYS.

LICENSE GRANT. SIG hereby grants to Licensee a nonexclusive license to install and use the Programs and Documentation as provided herein.

INSTALLATION AND USE. This license is a "single-user license"; it permits Licensee to install and use one copy of the Program at one time. If Licensee has multiple licenses for the Program, then Licensee may use as many copies of the Program at one time as it has licenses. Licensee is responsible for limiting the number of concurrent users to the number licensed. "Use" means a copy of the Program has been loaded and executed by any of the resident operating systems on a computer, and is in an active state. In an active state, the Program is capable of one or more of, but not limited to, the following: performing calculations, data processing, and/or data collection tasks in the background or foreground, displaying data or measurements derived from data in the foreground, and

saving results of data processing or display measurements to temporary or permanent computer memory or file.

Because this license permits use of only one copy of the Program at a time, Licensee is not allowed to distribute, or allow to be distributed, copies of the Program to other entities, colleagues, or locations which might contribute to or create a situation where it is not reasonably practicable to verify that only one copy of the Program is in use at one time.

Licensee shall use the Program only for its internal operations. "Internal operations" shall include use of the Program by Licensee's own employees or those of its subsidiaries or parent company, and in the performance of consulting or research for third parties who engage Licensee as an employee or independent contractor. Licensee may not allow use of the Program by colleagues, consultants, contractors, or students not employed or contracted by Licensee.

Licensee may make copies of the Program, including online documentation, only for backup or archival purposes. All copies of Program shall contain all copyright and proprietary notices in the original. "Online" means documentation that is

accessible in electronic form from the computer on which Program is currently active, either before, during, or after use of the Program.

Licensee is permitted to make one copy of the printed documentation provided with the Program for backup or archival purposes. Said copy of printed Documentation shall contain all copyright and proprietary notices in the original. Online documentation provided with the Program is subject to the same copy restrictions as the Program (see above).

Licensee shall not re-compile, translate, or convert source code examples distributed with the Program and Documentation for use with any non-SIG software. Licensee shall not attempt to obtain source code for any portions of the Program not provided in source code form, in whole or in part, by reverse engineering, decompilation, or disassembly.

Programs licensed to degree-granting educational institutes at SIG's educational discount are further restricted to use in connection with on-campus computing facilities that are used solely in support of classroom instruction and research activities of students and teaching faculty. The right to use the Program for commercial purposes is expressly excluded.

Licensee shall take appropriate action by instruction, agreement, or otherwise with any persons permitted to access to the Program, so as to enable Licensee to satisfy its obligations under this Agreement.

U.S. GOVERNMENT. If Licensee is acquiring the Program on behalf of any unit or agency of the U.S. Government, the following shall apply: (a) **For units of the Department of Defense: RESTRICTED RIGHTS LEGEND**: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause of DFARS 252.227-7013. (b) **For any other unit or agency: NOTICE**: Notwithstanding any other lease or license agreement that may pertain to, or accompany the delivery of the computer software and accompanying documentation, the rights of the Government regarding its use, reproduction and disclosure are as set forth in Clause 52.227-19(c)(2) of the FAR. Contractor/Licensor is Signalogic, Inc., 9704 Skillman #111, Dallas, TX 75243.

PAYMENT. Payment is due in United States currency within thirty (30) days of receipt of Program. Absent appropriate

exemption certificate(s), Licensee shall pay any taxes.

TERM OF AGREEMENT. This Agreement shall continue until terminated by SIG or Licensee as provided below.

TERMINATION. SIG may terminate this license by written notice to Licensee if Licensee (a) breaches any material term of this Agreement, (b) fails to pay the amount charged for this license within thirty (30) days after the date due, or (c) ceases conducting business in the normal course, becomes insolvent or bankrupt, or avails itself or becomes subject to any proceedings pertaining to insolvency or protection of creditors. Licensee may terminate this Agreement at any time by written notice to SIG. Licensee shall not be entitled to any refund if this Agreement is terminated, except of license fees paid for any Licensed Product for which the trial period has not expired at the time of termination. Upon termination, Licensee shall promptly return all copies of the Program and Documentation in Licensee's possession or control, or promptly provide written certification of their destruction.

LIMITED WARRANTY; LIMITATION OF REMEDIES. For a period of ninety (90) days from delivery, SIG warrants that each Program will conform in all material respects to the

description of such Program's operation in the Documentation. In the event that the Program does not materially operate as warranted, Licensee's exclusive remedy and SIG's sole liability under this warranty shall be the correction or workaround by SIG of major defects within a reasonable time. THE FOREGOING WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SIG SHALL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION LOST PROFITS. Licensee accepts responsibility for its use of the Program and the results obtained therefrom.

LIMITATION OF LIABILITY. AT NO TIME, FOR ANY REASON, SHALL SIG SHALL BE LIABLE TO LICENSEE FOR MORE THAN THE AMOUNT PAID BY LICENSEE TO SIG FOR THE PROGRAM.

GENERAL PROVISIONS. Licensee may not assign this License without written consent of SIG, except to a subsidiary or parent company of Licensee. Should any act of Licensee purport to create a claim, lien, or encumbrance on any Program,

such claim, lien, or encumbrance shall be void. All provisions regarding warranty, liability and limits thereon, and protection of proprietary rights, shall survive termination of this Agreement, as shall all provisions regarding payment of amounts due at the time of termination. This Agreement shall be governed by the laws of the state of Texas. Should Licensee install the Programs outside the United States, Licensee shall comply fully with all applicable laws and regulations relating to export of technical data. This Agreement contains the entire understanding of the parties and may be modified only by written instrument signed by both parties.

3.1.2 Media and Disk Space Requirements

NOTE: The media and disk space requirements listed below are in addition to the requirements detailed in section 1.3, System Requirements.

Hypersignal-Macro Series Media and Disk Space Requirements

1.44 Mbyte (3.5") floppy drive, 6 Mbyte hard disk space (4.5 Mbyte for Hypersignal-Macro NRT)

Related Topics

Hypersignal-Macro Series Media and Disk Space Requirements
DSP Source Code Media and Disk Space Requirements
DSPower-HWLib Media and Disk Space Requirements
DSPower-Block Diagram Media and Disk Space Requirements
Hypersignal-Macro Series Documentation and Media Inventory
DSPower-HWLib Documentation and Media Inventory
DSP Source Code Interface Documentation and Media Inventory
DSPower-Block Diagram Documentation and Media Inventory

DSP Source Code Media and Disk Space Requirements

1.44 Mbyte (3.5") floppy drive, 1/2 Mbyte hard disk space (does not include DSP device asm/linker/sw sim and/or C compiler, which could easily consume another 3-10 Mbyte of space)

Related Topics

Hypersignal-Macro Series Media and Disk Space Requirements
DSP Source Code Media and Disk Space Requirements
DSPower-HWLib Media and Disk Space Requirements
DSPower-Block Diagram Media and Disk Space Requirements
Hypersignal-Macro Series Documentation and Media Inventory
DSPower-HWLib Documentation and Media Inventory
DSP Source Code Interface Documentation and Media Inventory
DSPower-Block Diagram Documentation and Media Inventory

DSPower-HWLib Media and Disk Space Requirements

1.44 Mbyte (3.5") floppy drive, 2 Mbyte hard disk space

Related Topics

Hypersignal-Macro Series Media and Disk Space Requirements

DSP Source Code Media and Disk Space Requirements

DSPower-HWLib Media and Disk Space Requirements

DSPower-Block Diagram Media and Disk Space Requirements

Hypersignal-Macro Series Documentation and Media Inventory

DSPower-HWLib Documentation and Media Inventory

DSP Source Code Interface Documentation and Media Inventory

DSPower-Block Diagram Documentation and Media Inventory

DSPower-Block Diagram Media and Disk Space Requirements

1.44 Mbyte (3.5") floppy drive, 4 Mbyte hard disk space

Related Topics

Hypersignal-Macro Series Media and Disk Space Requirements

DSP Source Code Media and Disk Space Requirements

DSPower-HWLib Media and Disk Space Requirements

DSPower-Block Diagram Media and Disk Space Requirements

Hypersignal-Macro Series Documentation and Media Inventory

DSPower-HWLib Documentation and Media Inventory

DSP Source Code Interface Documentation and Media Inventory

DSPower-Block Diagram Documentation and Media Inventory

3.1.3 Documentation and Media You Should Have Received

This section contains an inventory of major items you should have received for different Signalogic DSP Software components. Choose from the list below.

Signalogic DSP Software Component Inventory

Hypersignal-Macro Series Documentation and Media Inventory

DSPower-HWLib Documentation and Media Inventory

DSP Source Code Interface Documentation and Media Inventory

DSPower-Block Diagram Documentation and Media Inventory

Hypersignal-Macro Series Documentation and Media Inventory

For all Hypersignal-Macro series packages, you should have received the following documentation: Users Manual, Tutorial, Software Reference Guide, and Hardware Reference Guide.

This documentation is delivered either as a set of four (4) books, or a set of three (3) books with Software Reference Guide and Hardware Reference Guide combined with a separator tab.

For all Hypersignal-Macro series packages, you should have received three 1.44 Mbyte diskettes. The first two diskettes contain executable, overlay, font, macro

function, utility, driver, and other program support files, and the third diskette contains example waveform files.

Related Topics

Hypersignal-Macro Series Media and Disk Space Requirements

DSP Source Code Media and Disk Space Requirements

DSPower-HWLib Media and Disk Space Requirements

DSPower-Block Diagram Media and Disk Space Requirements

Hypersignal-Macro Series Documentation and Media Inventory

DSPower-HWLib Documentation and Media Inventory

DSP Source Code Interface Documentation and Media Inventory

DSPower-Block Diagram Documentation and Media Inventory

DSPower-HWLib Documentation and Media Inventory

For the DSPower-HWLib software package, you should have received the Signalogic DSP Software System Guide, Users Guide, and Reference Guide.

This documentation is delivered in both online Help and printed book form. In addition, if you are using a Hypersignal package as a DSP Engine with DSPower-HWLib, then you should have received Hypersignal documentation. See Hypersignal-Macro Series Documentation and Media, above.

For the DSPower-HWLib software package, you should have received two (2) 1.44 Mbyte floppy diskettes.

Related Topics

Hypersignal-Macro Series Media and Disk Space Requirements
DSP Source Code Media and Disk Space Requirements
DSPower-HWLib Media and Disk Space Requirements
DSPower-Block Diagram Media and Disk Space Requirements
Hypersignal-Macro Series Documentation and Media Inventory
DSPower-HWLib Documentation and Media Inventory
DSP Source Code Interface Documentation and Media Inventory
DSPower-Block Diagram Documentation and Media Inventory

DSP Source Code Interface Documentation and Media Inventory

For a DSP Source Code Interface package, you should already have received Hypersignal documentation. Chapter 4 of the Hypersignal-Macro Series Hardware Reference Guide covers all DSP Source Code Interfaces, including ADSP-21xx, TMS320C2x, TMS320C3x, TMS320C4, TMS320C5x, DSP5600x, DSP9600x, DSP32C, DSP3210, and NEC77220.

Media for a DSP Source Code Interface package consists of one (1) 1.44 Mbyte floppy diskette.

Related Topics

Hypersignal-Macro Series Media and Disk Space Requirements
DSP Source Code Media and Disk Space Requirements
DSPower-HWLib Media and Disk Space Requirements
DSPower-Block Diagram Media and Disk Space Requirements
Hypersignal-Macro Series Documentation and Media Inventory
DSPower-HWLib Documentation and Media Inventory

DSP Source Code Interface Documentation and Media Inventory

DSPower-Block Diagram Documentation and Media Inventory

DSPower-Block Diagram Documentation and Media Inventory

For the DSPower-Block Diagram software package, you should have received the Signalogic DSP Software System Guide, Users Guide, Reference Guide, and Tutorial.

This documentation is delivered in both online Help and printed book form. In addition, if you are using a Hypersignal package as a DSP Engine with DSPower-Block Diagram, then you should have received Hypersignal documentation. See Hypersignal-Macro Series Documentation and Media, above.

For the DSPower-Block Diagram software package, you should have received two (2) 1.44 Mbyte floppy diskettes.

Related Topics

Hypersignal-Macro Series Media and Disk Space Requirements

DSP Source Code Media and Disk Space Requirements

DSPower-HWLib Media and Disk Space Requirements

DSPower-Block Diagram Media and Disk Space Requirements

Hypersignal-Macro Series Documentation and Media Inventory

DSPower-HWLib Documentation and Media Inventory

DSP Source Code Interface Documentation and Media Inventory

DSPower-Block Diagram Documentation and Media Inventory

3.2 Installing Signalogic DSP Software

This section covers installation, configuration, and checkout of Signalogic DSP Software components. For information on using software beyond initial configuration, see the Signalogic DSP Software Users Guide, or search select one of the main Users Guide topics below:

Main Users Guide Topics

Using Hypersignal-Macro Series Software

Using DSP Source Code Interface Software

Using DSPower-HWLib and DSPower-SPLib Software

Using DSPower-Block Diagram Software

Using Engine Manager

Using Hardware Manager

3.2.1 Installing Hypersignal-Macro Series Software

If you are installing a Hypersignal-Macro series software package *only*, then you can follow the instructions below. If you are installing a Hypersignal-Macro series software package *and* a DSPower series software package, please skip to Installing DSPower Series Software, below.

When installing a Hypersignal-Macro series software package only, a simple installation method

is described in Hypersignal-Macro Series Users Manual section 2.1, Installing Hypersignal. The quick form of this installation consists of these steps:

1. Copy *.* from distribution diskettes #1 and #2 to a subdirectory "X:\hsmacro", "X:\hsacous", "X:\hsmacnrt", etc.

2. Copy *.* from distribution diskette #3 to a subdirectory "X:\hsmacro\waveform", "X:\hsacous\waveform", etc. ("X" is an appropriate drive identifier for your system, "C", "D", etc.)

3. Run the appropriate batch file "hsm", "hsa", "hs", etc. at least once to ensure all files are expanded. See in section 3.2.5,

4. If you will be using DSP/analog hardware which has a memory-mapped transfer interface to the host PC, your system almost certainly will need the memory-mapped area to be "reserved" using the EMM386.EXE line in the system's CONFIG.SYS file. This is necessary to avoid conflicts with memory-resident software programs and utilities, as well as other installed hardware. For more information about this, see section 1.4.4, System Drivers and Resident Program Considerations, and EMM386.EXE Considerations (Reserving Memory Areas) in section 4.3.2, Memory-Mapped Transfer Gotchas.

NOTE: The Hypersignal-Macro Uses Manual may suggest an "X:\waveform" subdirectory for diskette #3. Use "X:\hsmacro\waveform", "X:\hsacous\waveform", etc. as explained above, instead.

For a thorough overview of Hypersignal-Macro series software installation and configuration, see Hypersignal-Macro Series Users Manual sections 2.1 through 2.5.3, or section 3.2.5, Installing and Configuring Hypersignal Software For User Mode Operation.

Related Installation Topics

Installing Hypersignal-Macro Series Software

Installing DSPower Series Software

Configuration and Checkout of DSPower-HWLib Software

Configuration and Checkout of DSPower-Block Diagram Software

Installing and Configuring Hypersignal Software For User Mode Operation

3.2.2 Installing DSPower Series Software

To install a DSPower series software package, follow the steps below:

1. Put diskette #1 into a 1.44 Mbyte (3.5") floppy drive. If you are installing both a DSPower-series package and a Hypersignal-Macro series package at the same time, *insert DSPower diskette #1*, and *not* Hypersignal diskette #1.

2. Select Program Manager | File | Run. If you are not yet familiar with the "|" notation, this means: "from the Program Manager window, pull down the 'File' menu selection, and choose 'Run').

3. Enter (using keyboard)

a:\setup <Enter>

or

b:\setup <Enter>

where choice of "a" or "b" depends on the configuration of your system's 1.44 Mbyte disk drive.

4. After the Signalogic Software Setup program initializes, you should see an "Install Options" dialog box displayed with checkboxes indicating which Signalogic DSP Software components will be installed. Make sure the checkboxes match the software packages you will be installing. If not, or you are absolutely certain that you already

have a current version of a package or packages installed correctly, then you can choose to uncheck one or more of the install options.

Important Note Depending on the Signalogic DSP Software components you have received, the installation checkboxes may also indicate a Hypersignal-Macro series software package. Although not necessary, Hypersignal-Macro series software packages can be installed and configured separately, using a simple installation method described in Hypersignal-Macro Series Users Manual section 2.1, Installing Hypersignal, and in section 3.2.5 below, Installing and Configuring Hypersignal-Macro Series Software for User Mode Operation.

5. After clicking OK on the Install Options dialog box, the Software Setup utility will prompt to verify the default subdirectory for installing files. Defaults are "C:\dspower" for DSPower series software packages and "C:\hsmacro" or "C:\hsacous" for Hypersignal-Macro series software packages. After you either affirm the default or enter a desired subdirectory, the Software Setup utility will begin installing software.

Important Note: You can specify a different subdirectory if desired, but it *must be located at the root level*, and you must *not change the names of subdirectories beneath* this choice. Doing so will cause undefined results in using Signalogic DSP Software products.

During installation, the Software Setup Utility may ask you to insert additional disks depending on the product(s) being installed. It may even ask you to re-insert a diskette that has already been used.

Important Note: If, during installation, you should see a box saying "file to be installed is older than existing file; keep existing file?", and showing statistical data from two different files with the same filename, you should select "keep the existing file". What has probably happened is a common library or DLL used by many software packages and needed by DSPower series software (such as a Borland or Microsoft control library) already exists on your hard disk drive. In such cases, this type of dialog box is displayed if the existing version is actually newer than the version the Signalogic Software Setup Utility is trying to install. This can happen if you have obtained a more recent copy of the common library by upgrading a software package, buying a new system, etc.

6. When finished, the Software Setup utility will display a final dialog box indicating successful installation. If you have installed a DSPower series package, this box will also prompt you to restart Windows. Answer yes to this question in order to allow the "Pipe.386" virtual device driver (VxD) to be made active, and the installation to fully take effect. *This is necessary for correct operation of DSPower-Block Diagram or DSPower-HWLib software when using a Hypersignal-Macro series software package as a "DSP Engine".*

If there are other applications or programs still open, then click on the "No" button, close these applications, and then exit and restart Windows manually. If the Pipe.386 VxD is not installed when you attempt to run a DSPower series package using Hypersignal software as a DSP Engine, you will get a message at some point indicating "pipe.386 VxD not loaded".

7. After software is installed, and Windows is restarted, you should see a Signalogic DSP Software group in the Program Manager window. Select one of the icons! And proceed to one of the configuration and checkout sections listed below:

8. If you will be using DSP/analog hardware which has a memory-mapped transfer interface to the host PC, your system almost certainly will require the hardware's memory-mapped area to be "reserved" using the EMM386.EXE line in the system's CONFIG.SYS file. This is necessary to avoid conflicts with memory-resident software programs and utilities, as well as other installed hardware. For more information about this, see section 1.4.4, System Drivers and Resident Program Considerations, and EMM386.EXE Considerations (Reserving Memory Areas) in section 4.3.2, Memory-Mapped Transfer Gotchas.

Related Installation Topics

Installing Hypersignal-Macro Series Software

Installing DSPower Series Software

Configuration and Checkout of DSPower-HWLib Software

Configuration and Checkout of DSPower-Block Diagram Software

Installing and Configuring Hypersignal Software For User Mode Operation

3.2.3 Configuration and Checkout of DSPower-HWLib Software

If you have installed DSPower-HWLib software, perform the following checks to make sure initial configurations are valid:

Digital Oscilloscope Demo

1. From the Signalogic DSP Software group under Program Manager, select the "HWLib Digital Scope Demo" icon.

2. You should first see the Hardware Selector dialog box, which lists supported DSP/analog hardware. Using the listbox, select the hardware type you have currently installed in your system. Make sure the base I/O address, base memory address (if applicable), and other configuration parameters are correct. See Signalogic DSP Software Users Guide section 8.2.2, DSP/Analog Hardware Configuration, for more information about configuring DSP/analog hardware. See also the appropriate hardware section in chapter 3 of the Hypersignal-Macro Series Hardware Reference Guide, for more information about specific hardware types.

NOTE: If you will be using DSP/analog hardware which requires a memory base address (i.e. memory base address not listed as 'none' in the Hardware Selector), then your system almost certainly will require the hardware's memory-mapped area to be "reserved" using the EMM386.EXE line in the system's CONFIG.SYS file. This is necessary to avoid conflicts with memory-resident software programs and utilities, as well as other installed hardware. For more information about this, see section 1.4.4, System Drivers and Resident Program Considerations, and

EMM386.EXE Considerations (Reserving Memory Areas) in section 4.3.2, Memory-Mapped Transfer Gotchas.

3. A "HWLib DLL" status box should appear, and you should see status messages indicating the progress of the DScope demo initialization process. The demo first tries to open Hypersignal-Macro as a DSP Engine; if that fails, it tries to open Hypersignal-Acoustic as a DSP Engine. Upon success in either case, it attempts to do the following:

1. initialize the hardware previously selected in the Hardware Selector dialog box
2. download a default DSP executable file to the DSP hardware
3. initializes the digital oscilloscope mode in the engine, and send down several parameters needed by the DSP code for operation
4. set up a window to use to display analog input data received from the engine, and also to receive "buffer ready" messages from the engine

If the digital oscilloscope demo fails to open a DSP Engine, you may need to recompile the dscope.cpp source code and put in an explicit engine path. You can do this in the second parameter of the DSEngineOpen call. See also Users Guide section 7.2.1, Opening DSP Engines.

4. If the selected DSP/analog hardware initializes correctly, you should see a rudimentary single-trace digital oscilloscope display, with the current channel 0 analog input (from the DSP/analog hardware) displayed along the horizontal axis. If you have a signal connected to the channel 0 input, you should see it (depending on the input signal amplitude and sampling rate being used, of course!).

5. The toolbar controls in the digital oscilloscope demo can be used to set vertical scale and position, display online Help, select an RMS indicator, etc. Moving the vertical scroll bar changes the current trigger level, which is represented by a horizontal blue marker (note that there is a small hysteresis region about the horizontal axis, where the trigger mode defaults to "freerun"). Moving the horizontal scroll bar changes the timebase, or amount of points displayed in each "frame" (update) of

the digital oscilloscope display. To change the sampling rate, push the toolbar "Fs" button.

6. You can change any aspect of the digital oscilloscope demo's operation by changing and recompiling the dscope.cpp source code file. Example project files to do this are provided for Borland C/C++ v3.1 (**.prj** filename extension), Borland C./C++ v4.5 (**.ide** filename extension), Microsoft Visual C/C++ v1.5 (**.mak** filename extension), and WATCOM C/C++^{32} v10.0 (**.wpj** filename extension). For detailed information, see Users Guide section 4.2, Creating Programs for DSPower-HWLib and DSPower-SPLib.

7. Consult the Signalogic DSP Software Users Guide for more information about the DLL status box (Users Guide section 7.5.3, Error/Status Window in Engine Manager), DLL calls being made in the dscope.cpp demo program (Users Guide section 7.3, DSPower-HWLib Function Overview), and the window procedure callback message structure (Users Guide section 7.4, Registering Callback Messages) needed to get and send data to and from DSP Engines and DSP/analog hardware.

Digital Tape Recorder Demo

1. From the Signalogic DSP Software group under Program Manager, select the "HWLib Digital Tape Demo" icon.

2. You should first see the Hardware Selector dialog box, which lists supported DSP/analog hardware. Using the listbox, select the hardware type you have currently installed in your system. Make sure the base I/O address, base memory address (if applicable), and other configuration parameters are correct. See Signalogic DSP Software Users Guide section 8.2.2, DSP/Analog Hardware Configuration, for more information about configuring DSP/analog hardware. See also the appropriate hardware section in chapter 3 of the Hypersignal-Macro Series Hardware Reference Guide, for more information about specific hardware types.

NOTE: If you will be using DSP/analog hardware which requires a memory base address (i.e. memory base address not listed as 'none' in the Hardware Selector), then your system almost certainly will require the hardware's memory-mapped area to be "reserved" using the

EMM386.EXE line in the system's CONFIG.SYS file. This is necessary to avoid conflicts with memory-resident software programs and utilities, as well as other installed hardware. For more information about this, see section 1.4.4, System Drivers and Resident Program Considerations, and EMM386.EXE Considerations (Reserving Memory Areas) in section 4.3.2, Memory-Mapped Transfer Gotchas.

3. A "HWLib DLL" status box should appear, and you should see status messages indicating the progress of the digital tape recorder demo initialization process. The demo first tries to open Hypersignal-Macro as a DSP Engine; if that fails, it tries to open Hypersignal-Acoustic as a DSP Engine. Upon success in either case, it attempts to do the following:

1. initialize the hardware previously selected in the Hardware Selector dialog box
2. set up a window to display controls needed for the demo, and also to receive update and status messages from the engine

4. If the selected DSP/analog hardware initializes correctly, you should see a digital tape recorder set of controls displayed, along with Vu meter, output waveform filename edit box, and sampling rate and sample counter indicators.

5. You can change any aspect of the digital tape recorder demo's operation by changing and recompiling the dtape.cpp source code file. Example project files to do this are provided for Borland C/C++ v3.1 (**.prj** filename extension), Borland C./C++ v4.5 (**.ide** filename extension), Microsoft Visual C/C++ v1.5 (**.mak** filename extension), and WATCOM C/C++^{32} v10.0 (**.wpj** filename extension). For detailed information, see Users Guide section 4.2, Creating Programs for DSPower-HWLib and DSPower-SPLib.

6. Consult the Signalogic DSP Software Users Guide for more information about the DLL status box (Users Guide section, 7.5.3 Error/Status Window in Engine Manager), DLL calls being made in the dtape.cpp demo program (Users Guide section 7.3, DSPower-HWLib

Function Overview), the window procedure callback message structure (Users Guide section 7.4, Registering Callback Messages) needed to receive update messages such as amplitude level and sample counter from DSP Engines and DSP/analog hardware, and the CONVERSIONINFO structure (Users Guide section 4.5, Acquiring Data Using DSPower-HWLib Software) required in the DSAcquireWvfrmFile and DSGenerateWvfrmFile calls used to implement the Record and Play controls.

Related Installation Topics

Installing Hypersignal-Macro Series Software

Installing DSPower Series Software

Configuration and Checkout of DSPower-HWLib Software

Configuration and Checkout of DSPower-Block Diagram Software

Installing and Configuring Hypersignal Software For User Mode Operation

3.2.4 Configuration and Checkout of DSPower-Block Diagram Software

If you have installed DSPower-Block Diagram, perform the following checks, to make sure initial configurations are valid:

1. If you have installed a Hypersignal-Macro series software package, select the top-level menu item "Configuration", and then "DSP Engines", and set the following:

    1. The "Hypersignal DSP Engine Command Line" should be set to the subdirectory, plus program name, which reflects where the Hypersignal package is installed; if not, set this now (e.g. C:\hsmacro\hsmacro.exe, C:\hsignal\hsmacnrt.exe, C:\hsacous\hsacous.exe, etc.). ***Do not use the user mode Hypersignal batch files*** (hsm.bat, hsa.bat, etc.).

2. The "Hypersignal Run-Time Source Path" should be the subdirectory on which the Hypersignal package is installed; if not, set this now (e.g. C:\hsmacro, C:\hsacous, etc.).

3. Under "Directories", which is also under the "Configuration" main menu item, ensure that "Waveform Files" is set to "c:\hsmacro\waveform", "c:\hsacous\waveform", etc.; if not, set this now.

2. If you will be using MATLAB as a DSP Engine, select the top-level menu item "Configuration", and then "DSP Engines", and set the following:

1. the "MATLAB DSP Engine Command Line" should be set to the subdirectory, plus program name, which reflects where MATLAB v4.x is installed; if not, set this now (for example, C:\matlab\bin\matlab.exe or similar).

2. the "MATLAB Run-Time Source Path" should be the subdirectory on which MATLAB is installed; if not, set this now (e.g. C:\matlab\dspower, C:\matlab\block, etc.). We suggest creating a new subdirectory under the MATLAB subdirectory for source files generated by DSPower-Block Diagram to ensure DSPower doesn't inadvertently write over existing .m files during its process

3. under "Directories", which is also under the "Configuration" main menu item, ensure that "Waveform Files" is set to "c:\matlab\waveform" or similar; if not, set this now. If you are also using a Hypersignal-Macro series package, you can leave this line set to "c:\hsmacro\waveform", "c:\hsacous\waveform", etc.

3. Select "DSP/Analog Hardware" (also under the "Configuration" main menu item), and set up any supported hardware that is installed in your system. To do this, select the hardware type you have currently installed in your system. Make sure the base I/O address, base memory address (if applicable), and other configuration parameters are correct. See Signalogic DSP Software Users Guide section 8.2.2, DSP/Analog Hardware Configuration, for more information about configuring DSP/analog hardware. See also the appropriate hardware section in chapter 3 of the Hypersignal-Macro Series Hardware Reference Guide, for more information about specific hardware types.

NOTE: If you will be using DSP/analog hardware which

requires a memory base address (i.e. memory base address not listed as 'none' in the Hardware Selector), then your system almost certainly will require the hardware's memory-mapped area to be "reserved" using the EMM386.EXE line in the system's CONFIG.SYS file. This is necessary to avoid conflicts with memory-resident software programs and utilities, as well as other installed hardware. For more information about this, see section 1.4.4, System Drivers and Resident Program Considerations, and EMM386.EXE Considerations (Reserving Memory Areas) in section 4.3.2, Memory-Mapped Transfer Gotchas.

Next, click on the "Set Hardware Variables" button, which will display a secondary, pop-up dialog box. Double-click on appropriate hardware variables on the lefthand side. ***If you only have one board installed in your system, then it's best to set all hardware variables to that board using "Set All".*** Note, however, that multitasking allowed by DSPower-Block Diagram allows more than one board to run and results to be displayed at the same time by setting up different hardware variables. See Signalogic DSP Software Users Guide section 8.2.2, DSP/Analog Hardware Configuration, for more information about configuring DSP/analog hardware using the Hardware Selector dialog box.

Related Installation Topics

Installing Hypersignal-Macro Series Software

Installing DSPower Series Software

Configuration and Checkout of DSPower-HWLib Software

Configuration and Checkout of DSPower-Block Diagram Software

3.2.5 Installing and Configuring Hypersignal Software For User Mode Operation

Installing and Configuring Hypersignal software is covered in full detail in Hypersignal-Macro Series Users Manual chapter 2, Getting Started, so this section and related topics are provided as a Windows online Help file version.

If you are installing a Hypersignal-Macro Series software package at the same time as a DSPower series package, then the Signalogic Software Setup utility will allow you to install both packages at once. And, if the Hypersignal package is being used as a DSP Engine by a DSPower series package, configuration of Hypersignal is handled automatically and transparently, which means you can probably ignore this section and related topics.

All Hypersignal-Macro series software packages can be operated in user mode or engine mode. This section deals exclusively with installing and configuring a Hypersignal-Macro series package for operation in user mode. For more information, see User Mode and Engine Mode in section 1.1,

Relationship of Signalogic DSP Software Components.

NOTE: Some of the sections listed below are essentially reprinted from Hypersignal-Macro Series Users Manual sections 2.1 through 2.5.3. Some sections may have material that is copyright © 1985-1990, by Hyperception, Inc. in addition to Signalogic's copyright © 1991-1995. All rights are reserved by their respective copyright holders.

Related Hypersignal Installation Topics

General Installation Topics

Installing Hypersignal-Macro Series Software

Installing DSPower Series Software

Configuration and Checkout of DSPower-HWLib Software

Configuration and Checkout of DSPower-Block Diagram Software

Installing and Configuring Hypersignal Software For User Mode Operation

Installing Hypersignal Without the Software Setup Utility

Below is the basic sequence for stand-alone installation of Hypersignal-Macro series software packages:

1. Copy *.* from distribution diskettes #1 and #2 to a hard disk subdirectory named "X:\hsmacro", "X:\hsacous", "X:\hsmacnrt", or

similar, depending on the Hypersignal package that you have ("X" is an appropriate disk drive identifier, "C", "D", etc.)

2. Copy *.* from distribution diskette #3 to a hard disk subdirectory "X:\hsmacro\waveform", "X:\hsacous\waveform", "X:hsmacnrt\waveform", etc.

3. Run Hypersignal using appropriate HSx.BAT file. See Running Hypersignal in User Mode from DOS.

NOTE: When installing Hypersignal software without the Software Setup Utility (i.e. using DOS command-line), the HSxxxxx.OV_ file (where "xxxxx"is macro, acous, macnrt, etc. depending on the Hypersignal version) file must be expanded, using the EXPAND.EXE utility, which is included with DOS version 5.0 and higher. When Hypersignal is run for the first time, after being installed as indicated above, the HSx.BAT files will do this for you automatically. The appropriate HSx.BAT file attempts to locate expand.exe from your system PATH list and on your \DOS and \WINDOWS subdirectories. See Running Hypersignal in User Mode from DOS for more information.

Running Hypersignal In User Mode from DOS

You can run Hypersignal-Macro series software in user mode by invoking the appropriate batch file provided with the software. These batch files first install the Metagraphics "MetaWindow" graphics driver (makes it memory-resident), runs Hypersignal, and

then remove the graphics driver from memory when you exit Hypersignal. This keeps the graphics driver out of system memory when you are not using Hypersignal.

NOTE: When a Hypersignal package is run transparently by DSPower-Block Diagram or DSPower-HWLib software as a DSP Engine, the MetaWindow graphics driver is not loaded or used in any way.

To run the appropriate Hypersignal-Macro series batch file, first select as the default (logged) subdirectory the "c:\hsmacro", "c:\hsacous", or or whatever name was used when installing the software (see section 3.2.5 above, Installing and Configuring Hypersignal for User Mode Operation) and run one of the following batch files: Below is a list of the correct batch files to use:

Table 11 Batch Files to Run Hypersignal in User Mode

	Hypersignal Package	

	Hypersignal-Macro NRT	
	Hypersignal-Macro or Hypersignal-Macro EX	
	Hypersignal-Acoustic	
	Hypersignal-Macro NRT, Nihon Version	
	Hypersignal-Macro or Hypersignal-Macro EX, Nihon Version	
	Hypersignal-Acoustic, Nihon	

Version	

To run one of the above batch files, enter its filename at the DOS command-line prompt, without the ".bat" extension, and push the 'Enter' key.

NOTE: When you run a Hypersignal-Macro series package in user mode for the first time, you will see a message of the form "HSxxxxx.CNF file not found -- initializing new configuration...". All this means is that you must set up driver selections and other basic information for the first time before the software is fully operational. Then, when you "exit" Hypersignal, all selections and configuration will be saved. When you next run the software, driver selection and other setup will not be required.

See Configuring Hypersignal in User Mode, below, for steps to follow when configuring a Hypersignal-Macro series package for the first time.

You can also invoke Hypersignal from a remote path. Generally, Hypersignal software searches for support files, its internal online Help file, and printer-driver definition file first on its own path, then on the currently logged DOS path, and then in the DOS path list.

The online Help and printer-driver definition files are also searched for on the path specified in the User Setup area (see below).

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from Windows

Configuring Hypersignal in User Mode

Math Coprocessor Usage in Hypersignal

Graphics and Mouse Driver Selection in Hypersignal User Mode

Graphics Driver Selection in Hypersignal User Mode

Graphics Driver Troubleshooting in Hypersignal User Mode

Mouse Driver Selection in Hypersignal User Mode

Mouse Driver Troubleshooting in Hypersignal User Mode

Configuration and Setup in Hypersignal User Mode

User Setup Area Initialization in Hypersignal User Mode

DSP/Analog Hardware Configuration in Hypersignal User Mode

External Program Configuration in Hypersignal User Mode

Running Hypersignal In User Mode from Windows

[@j DOS box, Hypersignal icon files, hs.ico, hsm.ico, hsa.ico]

If Windows 3.1x or higher is installed on your system, you can conveniently run Hypersignal-Macro series software packages from Windows by installing the Hypersignal icon (.ico) file included on distribution diskette #1. To do this, perform the following steps from Windows Program Manager:

1. Select Program Manager | File | New and select *program item.*

2. For **Description**, enter "Hypersignal-Macro" (or Hypersignal-Macro EX, Hypersignal-Acoustic, etc.); for **Command Line**, enter the path where the software is installed plus the appropriate batch filename as listed above in Running Hypersignal in User Mode from DOS (example: "c:\hsmacro\hsm.bat");, and for **Working Directory**, enter the installation path (same as Command Line but without batch filename). *Helpful hint: entering a space after the dash, as with "Hypersignal- Macro", may make the icon descriptor extend less to the sides, creating a "well-behaved program item" that doesn't bump up against it neighbors.*

3. Select the "Change Icon" button, and enter the installation path plus icon filename in the **Filename** box (example: "c:\hsacous\hsa.ico"). Icon filenames are hs.ico for Hypersignal-Macro NRT, hsm.ico for Hypersignal-Macro and Hypersignal-Macro EX, and hsa.ico for Hypersignal-Acoustic.

4. Upon return to Program Manager, you should see the new Hypersignal icon. After double-clicking, Hypersignal should run full-screen, with brisk performance, and all waveform displays and other graphics displays should be full-screen and fast. If this isn't the case, see section 4.1, Troubleshooting Installation Problems.

Use of the batch filename (hs.bat, hsm.bat, hsa.bat, etc.) as the command line program name is very important, because this automatically causes the correct .pif file to be loaded (hs.pif, hsm.pif, hsa.pif, etc.). These .pif files have been tailored to provide optimum performance for Hypersignal-Macro series software packages when run under Windows. See section 1.4.2, Performance Considerations, for more information about Hypersignal .pif files.

NOTE: If Hypersignal software is installed with the Software Installation Utility, a Windows program which is shipped standard with DSPower series software, icon

and .pif file setup is performed automatically.

If Hypersignal is executed in user mode from Windows, either from an installed icon or from a DOS prompt started from Windows, then it will be running in a "DOS box", which simulates a stand-alone DOS environment (also called real-mode). Actually, however, the host machine is in protected-mode, which is only possible with 386, 486 or higher processors. In protected mode under Windows, "time-slices" are still being given to Windows, even though it is in the background (not visible). Thus, the potential exists for Hypersignal software, and all its functions, displays, and hardware-based instrumentation, to run slower than in real-mode. See Hypersignal Performance Considerations, under section 1.4.2, Performance Considerations, for more information about operating Hypersignal in protected-mode.

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Configuring Hypersignal in User Mode

Math Coprocessor Usage in Hypersignal

Graphics and Mouse Driver Selection in Hypersignal User Mode

Graphics Driver Selection in Hypersignal User Mode

Graphics Driver Troubleshooting in Hypersignal User Mode

Mouse Driver Selection in Hypersignal User Mode

Mouse Driver Troubleshooting in Hypersignal User Mode

Configuration and Setup in Hypersignal User Mode
User Setup Area Initialization in Hypersignal User Mode
DSP/Analog Hardware Configuration in Hypersignal User Mode
External Program Configuration in Hypersignal User Mode

Configuring Hypersignal in User Mode

When you run a Hypersignal-Macro series package in user mode for the first time, either from DOS or from Windows, you will see a message of the form "HSxxxxx.CNF file not found -- initializing new configuration...". This only means you must make some initial driver and setup selections before the software is fully operational. Then, when you "exit" Hypersignal, all selections and configuration will be saved. When you next run the software, driver selection and other setup will not be required.

When running a Hypersignal-Macro series package in user mode for the first time (or in the event that the HSxxxxx.CNF file should be deleted), perform the following configuration steps:

1. Select a graphics driver (see Graphics Driver Selection in Hypersignal User Mode, for full discussion). Generally, using the "best-guess probe" value provided by the software will produce reasonable results--better resolution may be possible if you know exactly what kind of video board is in your system.

NOTE: When running a Hypersignal-Macro series software package under Windows, you may encounter problems with the currently installed Windows display driver in transition between Windows and Hypersignal full-screen display. See section 4.2, Troubleshooting Video and Display Problems, for more information.

2. Select a mouse driver (see Mouse Driver Selection in Hypersignal User Mode, for full discussion). Using the "best-guess probe" value provided by the Hypersignal software may or may not produce reasonable results. The probe accurately detects the presence of a memory-resident MOUSE.COM driver and some mice connected directly to the COM1 port, but has trouble with other mice and with COM2 port connections. Best results are almost always obtained if you have a Microsoft-compatible mouse connected to the COM1 or COM2 ports, and you enter '4C1', or '4C2', respectively.

NOTE: The following steps assume that you know how to make basic Hypersignal menu selections and enter data into Hypersignal menu fields. If not, please take a quick look at Hypersignal-Macro Series Users Manual section 3.1, Keyboard and Mouse Interface.

3. Set the subdirectory field in the User Setup menu selection to

"hsmacro\waveform" or "hsacous\waveform" (or whatever name you used for distribution diskette #3 files when installing the software).

4. If you have a real-time version of software (anything other than Hypersignal-Macro NRT), select the System Configuration menu item and enter the designator for your DSP/acquisition hardware (see DSP/Analog Hardware Configuration and Setup in Hypersignal User Mode, below, for full discussion) in the first four fields. The Hypersignal online Help for this menu item contains a full list of hardware designators. Selecting DSP/acquisition hardware is only necessary to run those Hypersignal functions requiring external analog input. Functions that can use the DSP as an accelerator will still operate using the 80x86 and/or 80x87 until a DSP board has been selected.

5. For both real-time and non-real-time software, enter the DEVELOPMENT DSP field in the System Configuration menu selection (last field, at the bottom of the menu). See the Hypersignal online Help for valid entries for this field. This field controls subsequent code generation and file acquisition from DSP manufacturer development tool output files. Entering this field is not required for the software to run--in fact, leaving it as a 'NONE' entry may be useful for DSP chip-independent

experimentation as you design your DSP algorithms.

6. At this point, you have completed basic Hypersignal user mode configuration. You can now run Hypersignal functions by entering filename fields with the example waveform filenames included on the distribution diskettes--example filenames include upchirp, speech, elliptic, and signal.

7. See Hypersignal Performance Considerations, under section 1.4.2, Performance Considerations, for information and suggestions concerning optimizing Hypersignal operation.

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Running Hypersignal In User Mode from Windows

Math Coprocessor Usage in Hypersignal

Graphics and Mouse Driver Selection in Hypersignal User Mode

Graphics Driver Selection in Hypersignal User Mode

Graphics Driver Troubleshooting in Hypersignal User Mode

Mouse Driver Selection in Hypersignal User Mode

Mouse Driver Troubleshooting in Hypersignal User Mode

Configuration and Setup in Hypersignal User Mode

User Setup Area Initialization in Hypersignal User Mode

DSP/Analog Hardware Configuration in Hypersignal User Mode

External Program Configuration in Hypersignal User Mode

Math Coprocessor Usage in Hypersignal

Regardless of whether operating in user mode or engine mode, all Hypersignal-Macro series software performs automatic detection and utilization of 80x87 series math coprocessors, if present in the host PC system. If a math coprocessor is present in your system, then the software should detect exactly which one (separate 8087, 80287, 80387 or onchip 80487 or higher), and prepare for use of the math coprocessor where appropriate in Hypersignal. If a math coprocessor is not present in your system, then all functions that require floating-point operations (such as Filter Construction, Difference Equations, and many others) are still performed with the same precision, but much slower (about 4 to 6 times slower). The decrease in speed occurs because software routines are used that exactly emulate the 80x87 math coprocessor. These software routines are much slower, because many 80x86 instructions are required to do what the 80x87 normally does in one.

If, because your system's 80x87 chip is faulty or for some other reason, you wish to override the automatic 80x87 detection then type at the DOS command-line (before executing Hypersignal):

(sets the DOS

environment variable "87" to OFF; forces the software to use 80x87 software emulation)

(sets the DOS environment variable "87" to ON; forces the software to use 80x87 hardware)

If you wish to re-activate automatic detection of the math coprocessor by the Hypersignal software, then type at the DOS command line (before executing Hypersignal):

```
(undefines the environment variable; returns software to automatic detection)
```

Most Hypersignal operations that require a high-degree of numerical precision use 8-byte precision. However, arbitrary filter design and some Difference Equation functions use 10-byte precision (also emulated exactly through software if an 80x87 is not present).

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Running Hypersignal In User Mode from Windows

Configuring Hypersignal in User Mode

Graphics and Mouse Driver Selection in Hypersignal User Mode

Graphics Driver Selection in Hypersignal User Mode

Graphics Driver Troubleshooting in Hypersignal User Mode
Mouse Driver Selection in Hypersignal User Mode
Mouse Driver Troubleshooting in Hypersignal User Mode
Configuration and Setup in Hypersignal User Mode
User Setup Area Initialization in Hypersignal User Mode
DSP/Analog Hardware Configuration in Hypersignal User Mode
External Program Configuration in Hypersignal User Mode

Graphics and Mouse Driver Selection in Hypersignal User Mode

NOTE: This section only applies if a Hypersignal-Macro series software package is being operated in user mode. This section does not apply if the Hypersignal software is being operated in engine mode, as is possible with DSPower-Block Diagram. For more information, see User Mode and Engine Mode, in section 1.1, Relationship of Signalogic DSP Software Components.

Upon executing Hypersignal-Macro series software for the first time, or selecting Graphics/Mouse Config from the main menu, two separate lists will be presented to you: first, a list of graphics drivers, and then a list of mouse drivers. If you have selected Graphics/Mouse Config from the main menu, then you can use the 'Esc' key to exit at any time, but if you are executing the software for the first time, you *must* choose graphics and mouse

drivers before the software will allow you do anything else.

When you next "exit" the software ('E' key from the main menu), the current driver selections are saved in the Hypersignal environment configuration save file, and no further graphics or mouse driver selection is required unless you choose the Graphics/Mouse Config selection from the main menu or receive a new version of software.

Hypersignal uses the Metagraphics "MetaWindow" memory-resident driver to provide direct support for over 100 different types of graphics hardware and 6 different types of mouse hardware. Supported graphics hardware includes CGA, EGA, VGA, PS/2, Hercules, AT&T, various portables, Wyse, and several popular 800x600 and 1024x768 formats. Supported mouse hardware includes Microsoft, Logitech, Mouse Systems, and others. Also supported directly is the NEC PC98xx series of machines and compatibles, which are widely used in Japan (the Nihon versions of Hypersignal-Macro, Hypersignal-Macro NRT, and Hypersignal-Acoustic are designed specifically to run on these machines). MetaWindow is manufactured by the Metagraphics Software Corporation--see the copyright notice at the beginning of the Hypersignal-Macro Series Users Manual.

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS
Running Hypersignal In User Mode from Windows
Configuring Hypersignal in User Mode
Math Coprocessor Usage in Hypersignal
Graphics Driver Selection in Hypersignal User Mode
Graphics Driver Troubleshooting in Hypersignal User Mode
Mouse Driver Selection in Hypersignal User Mode
Mouse Driver Troubleshooting in Hypersignal User Mode
Configuration and Setup in Hypersignal User Mode
User Setup Area Initialization in Hypersignal User Mode
DSP/Analog Hardware Configuration in Hypersignal User Mode
External Program Configuration in Hypersignal User Mode

Graphics Driver Selection in Hypersignal User Mode

NOTE: This section only applies if a Hypersignal-Macro series software package is being operated in user mode. This section does not apply if the Hypersignal software is being operated in engine mode, as is possible with DSPower-Block Diagram. For more information, see User Mode and Engine Mode, in section 1.1, Relationship of Signalogic DSP Software Components.

Hypersignal-Macro series software always performs a "probe" into your system hardware and returns a "best-guess" suggestion as to what

graphics driver choice should work. This probe is intended to serve only as a guide, and need not be followed in all cases. For example, if you had an 800x600 (SuperVGA) or 1024x768 resolution board in your system, the probe might return selection 42, which indicates standard 640x480 VGA hardware. Of course, this selection would work fine, but you could get better resolution by choosing the appropriate 800x600 or 1024x768 selection, depending on the manufacturer and model of the particular video board in your system.

NOTE: If you specify a color graphics driver (CGA, EGA, VGA, etc.) to be used with a monochromatic monitor, then add an "M" suffix to your selection; e.g. enter '42M' or '42m' for a VGA-compatible video card connected to a black and white monitor. Some monochromatic monitors, such as Compaq, will have excellent response to color displays, while others will not. The "M" suffix affects only menus, Hypersignal online Help, and other text displays, reducing the number of "color", or gray-level, values used in these displays.

If you have selected Graphics/Mouse Config from the main menu and end up making no change in the current graphics driver selection (a blank entry will use the current selection, which is displayed along with the "probe

best-guess" at the top of the display), then all waveform display settings will remain unchanged. However, after running Hypersignal for the first time, or after specifying a different graphics driver selection, all waveform display settings will be initialized to a default state. Altering only the monochromatic "M" parameter will not affect current waveform display settings.

NOTE: Don't forget the 'PgUp' and 'PgDn' keys! Over 100 different types of graphics driver selections are available. These are displayed in alphabetical order on 4 (four) different screen pages. If you don't see a certain driver selection at first, use the PgUp and PgDn keys to view all possible selections.

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Running Hypersignal In User Mode from Windows

Configuring Hypersignal in User Mode

Math Coprocessor Usage in Hypersignal

Graphics and Mouse Driver Selection in Hypersignal User Mode

Graphics Driver Troubleshooting in Hypersignal User Mode

Mouse Driver Selection in Hypersignal User Mode

Mouse Driver Troubleshooting in Hypersignal User Mode

Configuration and Setup in Hypersignal User Mode

User Setup Area Initialization in Hypersignal User Mode

DSP/Analog Hardware Configuration in Hypersignal User Mode

Graphics Driver Troubleshooting in Hypersignal User Mode

NOTE: This section only applies if a Hypersignal-Macro series software package is being operated in user mode. This section does not apply if the Hypersignal software is being operated in engine mode, as is possible with DSPower-Block Diagram. For more information, see User Mode and Engine Mode, in section 1.1, Relationship of Signalogic DSP Software Components.

If no graphics hardware is present in your machine, Hypersignal-Macro series software will display a message indicating that no hardware whatever could be found, and you will be unable to specify a driver selection. Basically, this condition will deactivate all waveform and instrumentation displays in the Hypersignal software. If you feel that this message is being displayed for your system incorrectly, then contact your software distributor immediately--there may be some sort of hardware or BIOS conflict that can be resolved.

If your graphics driver selection is not producing desired results, or you are unsure of just what type of video hardware is in your system, then the following discussion may help. As you may be aware, many

types of graphics hardware will support more than one type of driver listed. Also, some hardware may be available with varying amounts of memory. If displays appear to use only half the screen, or the display "wraps-around", try a driver selection for the same type of hardware, but with less vertical resolution. You may have to experiment to determine the choice that works best. If you are still unsure, try the following guesses--they represent some of the most commonly used types of hardware:

Table 12 Common Hypersignal-Macro Series Graphics Driver Selections

Host PC Type	Typical Graphics Driver Selection
	34 (CGA color), 32 (Hercules mono)
	37 (EGA color)

Host PC Type	Typical Graphics Driver Selection
	42 (VGA color), 65 (800x600 "autoswitch"), 76 (800x600 STB), 95 (800x6

Host PC Type	Typical Graphics Driver Selection
	00 VESA or 8514) 48 (VGA color), 46 (MCGA mono)

Some driver selections listed are for video hardware that use the "secondary display adapter" address as their default base address. Generally, such video hardware has some distinguishing characteristic about it that implies that it may be used in a system as the second video display, in

addition to a standard CGA, EGA, or VGA color display. Extremely high resolution (i.e. greater than 1200x1200) and monochromatic (black & white) operation are two common types of such hardware. When one of these boards is used in a system, it is easy to choose a driver selection that causes the wrong display base address to be used. Symptoms of a driver choice that assumes an incorrect base address include displays not being restored after striking the 'Esc' key, and unwanted text or strange "graphics" characters appearing in waveform display modes. If you are unable to get the software working correctly with non-standard graphics hardware that is listed as a driver selection, please contact your distributor, as problems of this nature can usually be resolved.

Some "compatible" machines do not always react to all combinations of text, graphics, and color as they should. For example, plasma or some LCD display machines must convert all color combinations to on, off, and possibly an intermediate shade. Usually they do this by maximizing contrast--black on white and white on black are translated equally well. However, sometimes the scheme used by the manufacturers does not always work flawlessly, especially in the case of colors that are not necessarily "dark" or "light". Try entering the 'M' monochromatic suffix if problems of this type should occur.

NOTE: Graphics driver selections for video hardware with 640x480 resolution or less typically require the use of one of the graphics font files provided on the Hypersignal distribution diskettes. These files have ".FNT" filename extensions. If no text is displayed in waveform displays or you think the font used is too small or too large, make sure that all .FNT files present on the original distribution floppy diskettes are in your working Hypersignal subdirectory. Graphics hardware with resolution greater than 640x480 generally use "bitmapped vector-generation" text, which requires no graphics font files.

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Running Hypersignal In User Mode from Windows

Configuring Hypersignal in User Mode

Math Coprocessor Usage in Hypersignal

Graphics and Mouse Driver Selection in Hypersignal User Mode

Graphics Driver Selection in Hypersignal User Mode

Mouse Driver Selection in Hypersignal User Mode

Mouse Driver Troubleshooting in Hypersignal User Mode

Configuration and Setup in Hypersignal User Mode

User Setup Area Initialization in Hypersignal User Mode

DSP/Analog Hardware Configuration in Hypersignal User Mode

External Program Configuration in Hypersignal User Mode

Mouse Driver Selection in Hypersignal User Mode

NOTE: This section only applies if a Hypersignal-Macro series software package is being operated in user mode. This section does not apply if the Hypersignal software is being operated in engine mode, as is possible with DSPower-Block Diagram. For more information, see User Mode and Engine Mode, in section 1.1, Relationship of Signalogic DSP Software Components.

Hypersignal-Macro series software always performs a "probe" into your system hardware and returns a "best-guess" suggestion as to what mouse driver choice should work. Even more so than the graphics driver, this probe is intended to serve only as a guide, and need not be followed in all cases. For example, the probe accurately detects the presence of a memory-resident MOUSE.COM driver and some mice connected directly to the COM1 port, but has trouble with other mice and with COM2 port connections. Best results are usually obtained by entering '4C1' or '4C2', which assumes a Microsoft-compatible mouse connected to either the COM1 or COM2 ports.

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Running Hypersignal In User Mode from Windows
Configuring Hypersignal in User Mode
Math Coprocessor Usage in Hypersignal
Graphics and Mouse Driver Selection in Hypersignal User Mode
Graphics Driver Selection in Hypersignal User Mode
Graphics Driver Troubleshooting in Hypersignal User Mode
Mouse Driver Troubleshooting in Hypersignal User Mode
Configuration and Setup in Hypersignal User Mode
User Setup Area Initialization in Hypersignal User Mode
DSP/Analog Hardware Configuration in Hypersignal User Mode
External Program Configuration in Hypersignal User Mode

Mouse Driver Troubleshooting in Hypersignal User Mode

NOTE: This section only applies if a Hypersignal-Macro series software package is being operated in user mode. This section does not apply if the Hypersignal software is being operated in engine mode, as is possible with DSPower-Block Diagram. For more information, see User Mode and Engine Mode, in section 1.1, Relationship of Signalogic DSP Software Components.

When using Hypersignal-Macro series software, after installing the MOUSE.COM driver, you will probably not be able to go back to operating the mouse directly through the COM1 or COM2 ports without first turning your machine all the way off (not just resetting or

rebooting). This is a known problem with the Metagraphics driver, and has to do with the way the COM1 and COM2 serial ports are initialized.

When you move the mouse very rapidly, if menu selection or target window display seems to "wrap around" or actually go in the opposite direction, there are two suggestions:

1. Use the '4C1' or '4C2' selections in the Hypersignal Graphics/Mouse Config menu.
2. If you are using selection '5', which instructs Hypersignal to use the installed mouse.com driver, you may be able to mitigate this effect by adjusting "speed" parameters for the driver. Most mouse.com drivers allow you to control the maximum rate of interrupts to application software using parameters on the "mouse" command line (usually found in the system's autoexec.bat file); lowering the rate of interrupts may help with any rapid mouse movement problem.

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Running Hypersignal In User Mode from Windows

Configuring Hypersignal in User Mode

Math Coprocessor Usage in Hypersignal

Graphics and Mouse Driver Selection in Hypersignal User Mode

Graphics Driver Selection in Hypersignal User Mode

Graphics Driver Troubleshooting in Hypersignal User Mode

Mouse Driver Selection in Hypersignal User Mode

Configuration and Setup in Hypersignal User Mode

User Setup Area Initialization in Hypersignal User Mode

DSP/Analog Hardware Configuration in Hypersignal User Mode

External Program Configuration in Hypersignal User Mode

Configuration and Setup in Hypersignal User Mode

NOTE: This section is valid whether Hypersignal software is being operated in user mode or engine mode. Note, however, that in engine mode, entries can be made only from macro language programs or external applications. For more information, see User Mode and Engine Mode, in section 1.1, Relationship of Signalogic DSP Software Components.

This section assumes that a Hypersignal-Macro series software package is already up and running, and you know how to choose menu items and enter data in menu fields using the data-entry editor. If not, please skim Hypersignal-Macro Series Users Manual section 3.3, The Hypersignal Main Menu, or Main Menu in Hypersignal, and section 3.4, Using the Data-Entry Editor, or Data-Entry Editor in Hypersignal, before continuing.

Basic software setup consists of making some initial selections in the Utilities main menu column. See the sections listed below for specific areas of configuration and setup:

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Running Hypersignal In User Mode from Windows

Configuring Hypersignal in User Mode
Math Coprocessor Usage in Hypersignal
Graphics and Mouse Driver Selection in Hypersignal User Mode
Graphics Driver Selection in Hypersignal User Mode
Graphics Driver Troubleshooting in Hypersignal User Mode
Mouse Driver Selection in Hypersignal User Mode
Mouse Driver Troubleshooting in Hypersignal User Mode
User Setup Area Initialization in Hypersignal User Mode
DSP/Analog Hardware Configuration in Hypersignal User Mode
External Program Configuration in Hypersignal User Mode

User Setup Area Initialization in Hypersignal User Mode

NOTE: This section is valid whether Hypersignal software is being operated in user mode or engine mode. Note, however, that in engine mode, entries can be made only from macro language programs or external applications. For more information, see User Mode and Engine Mode, in section 1.1, Relationship of Signalogic DSP Software Components.

Select the User Setup menu item. The drive field should already be set for you. The subdirectory field will be set to whatever was current when you first ran Hypersignal--you will want to change this to be your "hsmacro\waveform", "hsacous\waveform", etc. subdirectory. The TIME FILENAME and FREQ FILENAME fields don't really

matter, since you can enter these filenames in all functions anyway. If you don't like programs beeping and buzzing at you when you do something they don't like, then set the SOUND to 'OFF'. Everything else should be left at the default. Pressing the 'Esc' key or pressing the 'Enter' key (or pushing any mouse button) while the cursor is on the last field will leave this function.

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Running Hypersignal In User Mode from Windows

Configuring Hypersignal in User Mode

Math Coprocessor Usage in Hypersignal

Graphics and Mouse Driver Selection in Hypersignal User Mode

Graphics Driver Selection in Hypersignal User Mode

Graphics Driver Troubleshooting in Hypersignal User Mode

Mouse Driver Selection in Hypersignal User Mode

Mouse Driver Troubleshooting in Hypersignal User Mode

Configuration and Setup in Hypersignal User Mode

DSP/Analog Hardware Configuration in Hypersignal User Mode

External Program Configuration in Hypersignal User Mode

DSP/Analog Hardware Configuration in Hypersignal User Mode

NOTE: This section is valid whether Hypersignal software is being operated in user mode or engine mode. Note, however, that in engine mode, entries can be

made only from macro language programs or external applications. For more information, see User Mode and Engine Mode, in section 1.1, Relationship of Signalogic DSP Software Components.

Selecting DSP/Acquisition Board Drivers

Select the System Configuration menu item. If you have a real-time version of Hypersignal (any Hypersignal package besides Hypersignal-Macro NRT), then you should enter in the first four fields a designator for supported DSP/analog hardware. See the Hypersignal online Help for a list of designators for different manufacturers and board types.

NOTE: Typically, the first four fields of the System Configuration menu will be entered the same. There are only two cases where you would not have the first four fields all the same: you have two or more supported boards in the system, or you purposefully want to disable the DSP ACCELERATOR, in which case a selection of 'none' will cause the 80x86/80x87 to be used instead. A situation for the latter might be analog hardware that has no DSP chip, in which case either 'none' or some other DSP board selection would have to be entered for DSP ACCELERATOR.

ANALOG CONVERSION field

For real-time software versions only, this field selects the

DSP/acquisition hardware used by the Analog Conversion menu selection. See the Hypersignal online Help for a complete list of possible "hardware designators" that can be entered in this field. Default entry is 'NONE', which will cause access to the Analog Conversion function from the main menu to be denied.

For hardware with only A/D capability, D/A output will default to the DSP ACCELERATOR field entry (see below).

See the Hypersignal-Macro Series Hardware Reference Guide for a detailed discussion of individual boards, as well as hardware considerations such as method and maximum rate of data transfer, anti-aliasing, multichannel operation, and programmable hardware features.

DSP ACCELERATOR field

For real-time software versions only, this field selects the DSP hardware that will be used to accelerate DSP functions. See the Hypersignal online Help for a complete list of possible "hardware designators" that can be entered in this field.

Note that a few of the hardware designators listed in the Hypersignal online Help may not be valid, if they refer to "pure acquisition" hardware that has no onboard DSP device. Default entry is 'NONE', which will cause the host 80x86 processor and 80x87

coprocessor (if present) to be used instead.

During normal processing, a 'NONE' entry is perfectly valid, essentially deactivating the DSP accelerator, reducing processing to 80x86 and 80x87 speed levels, and making available the standard precision choices. In fact, when using fixed-point DSP hardware, the 'NONE' selection may be preferred for some types of FFT operations, such as filter impulse responses, where time required to process the input waveform is not a factor and high precision is needed.

NOTE: If you are using an analog-only board with no DSP chip, or no D/A playback capability, then whatever board you enter in the DSP ACCELERATOR field will assume those responsibilities. For example, if you are using an analog-only board for your SPECTRUM ANALYZER selection, then you must have a valid DSP ACCELERATOR selection working in order to run the spectrum analyzer, since the real-time Fourier analysis can't be done on your analog input board.

DIGITAL OSCILLOSCOPE field

For real-time software versions only, this field selects the DSP/acquisition hardware used by the Digital Oscilloscope function. See the Hypersignal online Help for a complete list of possible "hardware designators" that can be

entered in this field. Default entry is 'NONE', which will allow only waveform files as input (no analog input).

The DSP/acquisition hardware chosen must have multichannel capability in order to see a true dual-channel display, although different *dual-trace* display formats are possible from one analog input channel. See the Hypersignal-Macro Series Hardware Reference Guide for a detailed discussion of individual boards, as well as general discussion of various hardware considerations, such as sampling rates, triggering, real-time filtering, and DSP source code modification.

SPECTRUM ANALYZER field

For real-time software versions only, this field selects the DSP/acquisition hardware used by the Spectrum Analyzer function. See the Hypersignal online Help for a complete list of possible "hardware designators" that can be entered in this field. Default entry is 'NONE', which will allow only waveform files as input (no analog input).

The DSP/acquisition hardware chosen must have multichannel capability in order to see a true dual-channel display, although different *dual-trace* display formats are possible from one analog input channel. Intrachannel display options include simultaneous magnitude, phase, time, power spectra, and filtered vs. unfiltered.

See the Hypersignal-Macro Series Hardware Reference Guide for a detailed discussion of individual boards, as well as general discussion of various hardware considerations, such as sampling rates, triggering, real-time filtering, and DSP source code modification.

DSP Clock and External Sampling Clock Entry

After the four DSP/analog hardware function fields, an entry of the form '-NN' is possible to specify DSP clock rate. If such entry is made, NN should be a value in MHz, and is intended for DSP/acquisition boards that:

1. base their analog sampling rate on the main DSP clock, typically using an integer division, or submultiple method
2. are available in more than one clock rate version

For example, some TMS320C25-based hardware come in more than one version, with different DSP clock speeds; typical values might be 28, 40, or 50 MHz. Thus, you might enter 'XXXXX-40', where "XXXXX" is the hardware designator, to specify a 40 MHz version. If no clock rate entry is specified, the software will assume a default rate indicated in the "Performance Specifications" section of the hardware's entry in the Hypersignal-Macro Series Hardware Reference Guide. In most cases, the default is correct, since there are still relatively few boards that come in different speeds. For this reason, no entry (default) is highly recommended

unless you are absolutely sure of the DSP clock rate.

Also, an '-EXT' external clock "attribute" can be entered after any hardware designator in the above four fields, *where in this case "clock" is referring to the analog sampling rate clock, and not the DSP clock.* External clock entry deactivates any assumed sampling rate programming for the entered hardware, allows menu entry anywhere from 0 (dc) to 1 GHz in the Analog Conversion, Digital Oscilloscope, Spectrum Analyzer, and D/A playback-from-display functions, and passes a 16-bit or 32-bit integer value equal to the menu entry directly to the DSP source code. Do not attach this attribute for boards that do not truly have some sort of external clock input--otherwise they will "freeze up". Use this entry only if you are sure that the hardware's analog sampling circuitry is operating from an external clock. External clock entry can be made additionally to '-NN' entry described above.

MEMORY BASE ADDRESS and I/O BASE ADDRESS fields

These two fields, MEMORY BASE ADDRESS and I/O BASE ADDRESS, are somewhat more complicated. Normally, you will want to leave these two fields at their default (default values are listed in the Hypersignal online Help and in the Hypersignal-Macro Series Hardware Reference Guide). However, there can be times when

you need to change base addresses--the best examples being a possible conflict with boards you already have in your system, or a manufacturer that ships different boards with different base addresses.

Note that there is one entry for most manufacturers supported--not for every board. If you need to change a base address, *be sure you only change the value for the relevant manufacturer*--the other values do not need to be correct, but *they must remain entered!* Enter zeros for the other manufacturers if you must--but enter one value for every manufacturer listed in the field label (you can always get the defaults back by deleting the configuration file, if necessary).

If the manufacturer of your DSP/acquisition hardware does not appear on the menu, then you will have to change memory and I/O base addresses directly in the user-environment configuration file. See Hypersignal-Macro Series Users Manual section 3.7, Accessing and Updating the User-Environment Configuration File, or Configuration File in Hypersignal, and also the "config.doc" file, included on Hypersignal distribution diskette #2, for instructions on how to do this.

Multiple Boards From Same Manufacturer

If you have two or more boards in your system from the same

manufacturer, then Hypersignal will consider the first board mentioned in the four "operation" fields (ANALOG CONVERSION, DSP ACCELERATOR, DIGITAL OSCILLOSCOPE, and SPECTRUM ANALYZER) to be at the entered I/O base address, the second board to appear in the operation fields to be at the I/O base address plus 10H (16), the third board at the I/O base address plus 20H, and so on. Note that the software has no way of knowing that two boards in the system at the same time unless you have entered both designators at least once in the first four fields of the System Configuration menu. If only one is entered then the software assumes that this board is located at the corresponding base address value, regardless of how you have set the jumpers on the board.

NOTES: All memory and I/O base addresses must be entered in hexadecimal. Please consult the Hypersignal-Macro Series Hardware Reference Guide to see if your DSP/acquisition hardware requires a base address entry other than the default before it will operate. Remember, some manufacturers send out different boards with different base addresses.

All memory and I/O addresses must not conflict when different boards are present in the system at the same time. If you change the base address jumper(s) on a board

from the factory configuration, you must also change the appropriate memory or I/O base address field.

The "Exit,Save" function ('E' key) from the main menu will save all entries for use in later sessions.

A few supported hardware manufacturers use a separate, memory-resident driver that requires base address setup to be done outside of Hypersignal (usually via a driver command-line switch or a DOS environment variable). Consult the appropriate section in the Hypersignal-Macro Series Hardware Reference Guide for your hardware to see if this is the case.

80286/80386 INSTRUCTION SET field

This field should generally be set to 'Y' if you have an 80286, 80386, or 80486 machine. Depending on the DSP/acquisition hardware design and data transfer scheme implemented by the hardware manufacturer, this field may or may not be relevant. Only change it to 'N' if you definitely have an 8086 or 8088 (IBM PC/XT machine) or you are specifically instructed to do so in the Hypersignal-Macro Series Hardware Reference Guide, depending on your particular DSP/acquisition hardware.

Symptoms of leaving this field set to a 'Y' when not appropriate for certain DSP/acquisition hardware include "spikes" in your analog

input data, or board freeze-ups during real-time operation.

General Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Running Hypersignal In User Mode from Windows

Configuring Hypersignal in User Mode

Math Coprocessor Usage in Hypersignal

Graphics and Mouse Driver Selection in Hypersignal User Mode

Graphics Driver Selection in Hypersignal User Mode

Graphics Driver Troubleshooting in Hypersignal User Mode

Mouse Driver Selection in Hypersignal User Mode

Mouse Driver Troubleshooting in Hypersignal User Mode

Configuration and Setup in Hypersignal User Mode

User Setup Area Initialization in Hypersignal User Mode

External Program Configuration in Hypersignal User Mode

External Program Configuration in Hypersignal User Mode

NOTE: This section is valid whether Hypersignal software is being operated in user mode or engine mode. Note, however, that in engine mode, entries can be made only from macro language programs or external applications. For more information, see User Mode and Engine Mode, in section 1.1, Relationship of Signalogic DSP Software Components.

Hypersignal-Macro series software can be configured to execute

external programs from be selecting customized menu items. For example, you might want to set up a text-editor as a menu item under the Utilities column, so that you can go directly to your editor from Hypersignal to review generated source code, or set up a debugger so that you can see what Hypersignal has downloaded and is executing on your DSP board. To do this, select the Ext Program Config menu item. In the appropriate field for where you want the external program menu label to appear on the Hypersignal main menu, enter the filename of your program, followed by a comma, followed by the menu label as you would like it to appear. For instance, an editor might be set up by entering

```
editor,text-editor
```

in the UTIL FILE, MENU LABEL 1 field. Such an entry would cause Hypersignal to look for an executable file of name "editor.com" first, then "editor.exe", then "editor.bat", in that order, whenever the menu item "TEXT-EDITOR" is selected from the Hypersignal main menu. Of course, you can explicitly enter an extension if not ".COM", ".EXE", or ".BAT". Be sure to specify where the external file is located in the DRIVE and SUBDIRECTORY fields, although Hypersignal will look elsewhere, including the DOS path list, for the file if not found immediately.

It is also possible to execute external programs from Hypersignal macro programs. This technique is used extensively by many of the additional functions included with Hypersignal-Macro EX and Hypersignal-Acoustic. The technique is also used extensively by DSPower-Block Diagram function files responsible for implementing and controlling Hypersignal blocks from the DSPower-Block Diagram environment.

Related Hypersignal Installation Topics

Running Hypersignal In User Mode from DOS

Running Hypersignal In User Mode from Windows

Configuring Hypersignal in User Mode

Math Coprocessor Usage in Hypersignal

Graphics and Mouse Driver Selection in Hypersignal User Mode

Graphics Driver Selection in Hypersignal User Mode

Graphics Driver Troubleshooting in Hypersignal User Mode

Mouse Driver Selection in Hypersignal User Mode

Mouse Driver Troubleshooting in Hypersignal User Mode

Configuration and Setup in Hypersignal User Mode

User Setup Area Initialization in Hypersignal User Mode

DSP/Analog Hardware Configuration in Hypersignal User Mode

3.3 Release Notes

Normally, the online Help documentation will be the most up-to-date information available for Signalogic DSP Software

components, except in cases where features already documented did not make it into the current release of software***. Be sure to check the *.doc files included with each software component to see if in fact there are features not yet installed or known bugs and/or workarounds in the current version.***

3.3.1 Revision History for Hypersignal-Macro Series Software

[@j release notes, new features, enhancements, bug-fixes, corrections]

Following is revision history of Hypersignal-Macro RT, Hypersignal-Macro EX, Hypersignal-Acoustic, and Hypersignal-Macro NRT software since June 1991. The history is organized in reverse chronological order, with most recent enhancements/corrections listed first.

NOTE: Use the Browse (">>" and "<<") buttons to scroll forward and backwards in the release notes.

Most Recent Release Notes

New Features, Hypersignal-Macro Series, MAR95

Bug-Fixes/Corrections, Hypersignal-Macro Series, MAR95

New Features, DSPower-HWLib, MAR95

Bug-Fixes/Corrections, DSPower-HWLib, MAR95

New Features, DSP Source Code Interface, MAR95
Bug-Fixes/Corrections, DSP Source Code Interface, MAR95
New Features, DSPower-Block Diagram, APR95
Bug-Fixes/Corrections, DSPower-Block Diagram, APR95

Release Notes, Hypersignal-Macro Series, MAR95

Release notes for Hypersignal-Macro series software from March 1995 through present cover Hypersignal-Macro RT and Hypersignal-Macro EX v4.00xf, Hypersignal-Acoustic v4.00xf, and Hypersignal-Macro NRT v5.00xf.

New Features, Hypersignal-Macro Series, MAR95

1) Automatic processing of either .WAV or .TIM files by lcorr, lconv, lfft, and wlt functions. .WAV can also be specified in output filenames when using these functions.

2) First version of support for Innovative Integration PC50 support added. See Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for notes.

3) User-defined window functions enabled in FFT Generation, LPC Autocorrelation, and Spectrum Analyzer. If one of the defined window names (e.g. 'hm', 'bt', 'r', etc.) is not entered, then the entry is assumed to be either a .tim or .wav waveform file. NOTE: the

window waveform length should be equal to the framesize entered. A suggestion to avoid mismatches, using FFT Generation as an example, would be to write a short macro function "FFT_Ex" or similar. This function would prompt for the same parameters as FFT Generation, and then call Difference Equations (or a user-defined module) to generate the exact window needed, before invoking FFT Generation.

4) 'BAR' entry allowed in Spectrum Analyzer TPL field, in addition to 'LIN' and 'PNT'. 'BAR' is now the default entry in Hypersignal-Acoustic; 1/N octave-band frequency axis display (LOG FREQ field) no longer automatically invokes bar-type display if 'LIN' or 'PNT' entry is made in TPL field.

5) Output attenuation list values added to Analog Conversion function for multichannel boards. Tested initially with Sig32C-8 board. See Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for detailed information.

6) Full multichannel implementation completed for Innovative Integration PC31 board. All eight (8) input channels, including multiplexing, and all four (4) DAC channels, are operational from software menus.

7) Addition of OpenTextFile, ReadTextFile, and CloseTextFile commands to macro language. See

Hypersignal-Macro Series software addendum.doc file (distribution diskette #2).

8) Addition of SetWaveformFileMode and SetTextFileMode commands to macro language; 0 is default mode (as before), 1 specifies that file errors should not cause a macro abort. When an error occurs in this mode, file functions (GetHeaderValue, OpenTextFile, etc.) typically return -1.

9) Addition of '==' and '!=' comparisons, and addition of '!' and '~' operators to macro language. Addition of "boolean" type and TRUE and FALSE reserved constants to macro language. See Hypersignal-Macro Series software addendum.doc file (distribution diskette #2).

10) Addition of .WAV file read to real-time filter download functions; these functions now default to first .TIM, and then .WAV binary waveform files when .IIR and .FIR filter coefficient (text) files cannot be located.

11) Fully updated version of Hardware Reference Guide released; March 1995, revision 3.5.

12) Addition of long MLS lengths for DSP5600x boards, including TBS Tahiti and Techron TEF20HI. Previous MLS maximum length was from 4095 to 32767, depending on the DSP5600x hardware type and amount of external memory. New maximum length is 262,143. Hypersignal-

Acoustic only. NOTE: MLS lengths 8,191, 16,383, and 65,535 are not yet operational.

Bug-Fixes/Corrections, Hypersignal-Macro Series, MAR95

1) Correction to Convolution/Correlation function when specifying .WAV file input in the filter (impulse response) field.

2) Correction to gain list entry for last two channels when using Sig32C-8 board in Analog Conversion function.

3) Documentation of ProPort and MP3210 combination in online Help; correction to sampling rate programming for this combination. Enter 'ARI3210-B' in System Configuration menu.

4) Addition to driver for Innovative Integration to implement software workaround for dual-port memory bug on Rev. F and earlier PC31 boards.

5) Addition to driver for Turtle Beach System Tahiti board to implement consistent setting of digital potentiometer registers upon driver reset. Currently, the setting is the midpoint, which creates about a 13dB gain when output from the Tahiti is looped back to input.

6) Correction to .WAV format read functions where non-standard .WAV formats were not always handled correctly.

7) Correction to assembly language code generation errors in some DSP5600x IIR filters where rounding was being used, and where illegal instruction sequence was being generated.

8) Correction to Analog Conversion function to allow acquisition of waveform files with long framesize values and extended headers.

Release Notes, Hypersignal-Macro Series, NOV94-FEB95

Release notes for Hypersignal-Macro series software from November 1994 through February 1995 cover Hypersignal-Macro RT and Hypersignal-Macro EX V4.00xe, Hypersignal-Acoustic V4.00xe, and Hypersignal-Macro NRT 5.00xe.

New Features, Hypersignal-Macro Series, NOV94-FEB95

1) An "engine mode" command line parameter has been enabled, similar to driver mode (see below), but allowing Hypersignal to be invoked from DOS or Windows programs without a device driver being installed. As with driver mode, the standard Hypersignal graphics drivers are not loaded (freeing up about 130 kb of low memory), which causes display and instrument functions to be disabled. Also like driver mode, Hypersignal runs "quietly" and produces no

screen output of any type. Below are DOS command line and Windows C/C++ program examples to do this:

```
hsmacro /e fname.mac
```

```
hsacous /e fname.mac
```

```
StatusCode =
WinExec("C:\\hsignal\\h
smacro.pif /e
fname.mac", SW_HIDE);
```

```
StatusCode =
WinExec("C:\\hsacous\\h
sacous.pif /e
fname.mac", SW_HIDE);
```

In this mode, Hypersignal interrogates the system for an external device "pipe" driver, and if one is not found (or is not connected to another application), disables interactive communication, graphical output, error/status output, and all macro language non-file input/output. The macro language file specified by fname.mac should contain no display or instrument functions. Also, it should not use the $MANUAL or $WAITFORKEY commands, and it should include an Abort or Exit command to cause termination of the session. Note that the HSxxxx.bat files should not be used from the DOS command line or WinExec routine, or the graphics drivers will be installed unnecessarily.

2) First version of support added for Domain Tech. DSPCard TMS320C31 board.

Enter 'DOMC31' in System Configuration menu. The DSPCard contains a 32 MHz TMS320C31, 32k x 32 or 128k x 32 SRAM, and 14-bit TLC32044 AIC analog I/O channel. See the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for details on using this board, setting the correct clock rate entry in the System Config menu, and data-entry menu usage.

3) Support added for multiprocessor DSP boards. Enter '/N' after board designator and processor clock (if any), but before DSP download filename (if any), where N is processor number. Default processor number (no entry) is zero. See the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for more information.

4) Addition of SetEscMode and $SETESCMODE commands to macro language. See the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for more information.

5) Addition of "auto-configure" values in Analog Conversion D/A output.

Entering 0 for SAMPLING FREQUENCY and NUMBER OF SAMPLES fields, and null (blank) entry for the CHANNEL LIST field, will cause these values to be set using input waveform file header information. These options are especially useful for macro language operation, where it may

not be know in advance which waveform files will be sent to the D/A output.

Bug-Fixes/Corrections, Hypersignal-Macro Series, NOV94-FEB95

1) Correction to long FFT, convolution, and correlation functions when used with previously existing waveforms with extended headers.

2) Correction to D/A playback from 2-D Spectrograph when framesize = FFT size and no overlap was used. In such cases the 2-D Spectrograph function failed to play back the correct amount of waveform designated.

3) Correction to uninitialized variable which could cause the Digital Oscilloscope and Spectrum Analyzer functions to crash if used for first time with waveform files only as input (i.e. no hardware analog input).

4) Correction to Pole-Zero Plot function, which left a filter coefficient file open and could keep other functions from using this file if SHARE.EXE is installed in the host system.

5) Problem with D/A playback for TMS320C25 board from macro language files corrected. Boards that previously failed to respond correctly to a D/A playback call from a macro program included the Ariel DSP-16.

6) Additions to Innovative Integration PC31 Hypersignal driver and TMS320C31 DSP code to prevent output "pops" at the end of D/A operations, and to correct hang-up problems with machines that have external memory conflict with PC31 memory-mapped dual-port SRAM.

7) Correction to Ariel DSP-3210 driver in case where DSP-3210 board has two processors installed.

8) Correction to nested procedure calls inside macro language files.

9) Correction to Analog Conversion D/A looping ('GL' entry) for some lengths of waveform files involving buffer boundary conditions.

10) Correction to Difference Equations output for |equation output| <= 1 for sample lengths longer than 4086. Output was being truncated to +/- 1.

11) Correction to explicit clock entry in hardware designator fields of System Config menu. Previous entry was not using precision beyond 4 significant digits. Current precision is the full length of each entry.

Release Notes, Hypersignal-Macro Series, JUL94-OCT94

Release notes for Hypersignal-Macro series software from July 1994 through October 1994 cover Hypersignal-Macro RT and Hypersignal-Macro EX V4.00xd,

Hypersignal-Acoustic V4.00xd, and Hypersignal-Macro NRT 5.00xd.

New Features, Hypersignal-Macro Series, JUL94-OCT94

1) The HSx.bat (hsm.bat, hsa.bat, hs.bat, etc.) file attempts to expand the Hypersignal overlay file and other compressed files when Hypersignal is run for the first time after installation. See the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for more information.

2) First versions of very long FFT, convolution, and correlation are now included with HS-Macro EX. A beta version MLS stimulus & response function is now included with HS-Acoustic. These functions are implemented as protected-mode programs controlled by macro functions (i.e. use lfft, lconv, lcorr, and mls in the Macro Manager menu). The use of extended-DOS .exe programs allows previous Hypersignal memory limitations to be circumvented, and large amounts of memory to be utilized. You should have at least 4 Mbyte of extended memory in your system if you need to use these functions. These functions are included on disk 1 for HS-Macro EX and HS-Acoustic, along with the DOS extender program DOS4GW.exe. See the Hypersignal-Macro Series software addendum.doc file (distribution

diskette #2) for more information about these macro functions.

3) A beta-version "equalize" macro function has been added to HS-Acoustic. Combined with the Real-Time Convolver, this function can be used to equalize a transfer function measured in the HS-Acoustic Spectrum Analyzer, using either real-time analog input, or waveform file input. See the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for more information.

4) Hypersignal Magnitude Display, Phase Display, and 3-D Spectrogram have been modified to display long FFT sizes, and to display DFT sizes that are not necessarily a power of 2. Phase Display for long sizes is limited to wrapped phase only (i.e. unwrapped not enabled). 3-D Spectrogram display for long sizes does not include the "mesh" (cross-hatch) option. There are no limits in Magnitude Display. The 2-D Spectrograph and other frequency domain functions ARE NOT enabled for long FFT and DFT sizes.

5) First version of .WAV file support added. See the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for more information.

6) Full complex arithmetic added to Difference Equations function. Entry of 'j' is treated as sqrt(-1). Complex input waveforms are handled correctly, and complex

output is generated if needed. The following functions have been added to support this:

- Re(arg) and Im(arg) Take real and imag parts of complex value. If no other complex values are in the equation, then real-valued output is generated.
- Con(arg) Calculates conjugate of complex value.

Also, standard functions such as exp, sqr, abs, etc. have been modified to accept either real or complex values.

7) The frequency zoom macro function included with HS-Macro EX and HS-Acoustic has been updated. The new function is slower but more accurate. See comments inside fzoom.mac file.

8) The sampling rate conversion macro function has been renamed to fsconv and can also be called from the dsplib.inc library. Use fsconv.mac from the Macro Manager menu.

9) Optional entry of full-scale voltage for individual hardware types is now possible in the System Configuration menu. Entry of '/vv' after a designation is interpreted as vv Volts full-scale. For hardware that has software-selectable gain amplifiers, this value can be used should the input impedance of hardware be changed, so that the INPUT RANGE/GAIN fields in the instrument functions will reference the correct value to make gain setting determinations. For example, a hardware type with default full-scale voltage of 1.4 V is modified so that its full-scale

voltage is not 10 V. After entry of 'xxxxx/10' in the System Config menu (where xxxxx is the hardware designator), an entry of 1, v1, or d20 in an instrument INPUT RANGE/GAIN field would cause a gain of 20 dB to be set.

For hardware without hardware-programmable gain, or in cases where the input impedance of the hardware has not been changed, no entry full-scale voltage entry should be made.

10) Re-definition of gray-scale patterns used when plotting 2D contour plots to LaserJet and other B/W printers. Improved gray-scale effect.

11) Addition of 'ARI56-DM' designator in System Configuration menu to allow fully correct operation of the Ariel PC-56D and Digital Microphone combination. Hypersignal-Acoustic only.

12) First version of support for Innovation Integration PC31 TMS320C31 board installed. Use 'IIC31' in System Configuration menu. For processor clock values other than default 40 MHz, enter the clock value also (e.g. 'IIC31-33'). See the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for important, additional information.

13) StrCopy, StrVal, and StrLen functions added to macro language. StrCopy example:

```
s1 = StrCopy(s2, 2, 4)
```

causes 4 characters from string s2, starting at position 2, to be copied into string s1. StrVal example:

```
j = StrVal(s1)
```

converts the string s1 into the value j. The returned value is automatically type converted. StrLen example:

```
slen = StrLen(s1)
```

returns the length of the string s1, and stores into slen, which can be real or integer.

14) Macro language include file directive ({INC filename}) modified to search for filename using explicit path information if contained in filename, otherwise try first the path specified in the Macro Manager menu, and then the Hypersignal .exe path.

15) "autoinit" or "auto" option added to macro language menu field inputs for functions and displays. This option does the same thing as "no_change", but can be used to imply that the software is to fill in these fields at run-time, which can make macro files more readable.

16) Circular cross-correlation option added to Convolution/Correlation time domain menu function. Circular correlation proceeds such that input waveform data repeats when the end of the waveform is reached. Up to filt_len pts. of input waveform data is repeated, where

filt_len is the length of the filter being applied.

Bug-Fixes/Corrections, Hypersignal-Macro Series, JUL94-NOV94

1) Correction to inverse FFT function crash that resulted from broken code in release 4.00xc.

2) For Nat. Inst. AT-DSP2200 board: waiting period to remove artifacts due to digital filter settling time after ADC and DAC calibration. First 128 samples are discarded before acquisition / output / measurement / etc. event occurs.

3) Correction to problem where cursor movement before waveform display was finished caused erroneous input (for example, help screen to display). Only occurred with dual-display, and was only a problem in 4.00xc.

4) Correction to waveform editing problem where swapping trace names or moving trace names, without actually changing names, could cause errors between display-and-edit sessions.

5) Correction to D/A output from dual-display of two stereo waveforms.

6) Correction to Ariel DSP-C40 board automatic memory sizing, which failed to determine actual amount of Global SRAM in some cases. Additionally, an entry of ARIC40-B in System Config menu causes data buffers to be located in

DSP-C40 Global SRAM memory instead of default DRAM memory.

7) Correction to Recursive Filtering when input waveform file header contains a FRAMESIZE value of zero.

8) Correction to Waveform Editing in Point Mode, with stereo (2-channel) waveforms and SHARE.EXE installed, in which case point mode changes sometimes didn't "stick" or caused the display to be kicked back to the data-entry menu.

9) Correction to comparison of null strings in macro language if-then statements.

10) The frame multiplier in Difference Equations was not lowest-precedence among operators, as it should be. The online Help was updated to reflect this correction. This change is not yet present in the Users Manual, in which section 4.6.1 talks about a correlation example using the frame multiplier. The expression for the pattern waveform in this example should be changed to x2[0,m], and the comments about precedence ignored.

11) Gaussian window choice in FFT Generation was not displayed correctly in frequency domain displays.

12) Frequency zoom fzoom (macro) function corrected to truncate excess, zero-valued length of final output waveform.

13) Correction made to point-mode editing in Waveform Display/Edit function, where initial point-mode edit could result in an invalid output waveform header if a region mode editing operation had not been performed first.

14) Problem corrected in iFFT Generation function, where differences between .REA (real) and .IMA (imaginary) waveform file headers were not being correctly reconciled. Some legitimate differences can occur if the .REA and .IMA files are generated independently, for example using Difference Equations.

15) Access problem that could occur when share.exe is installed, depending on the sequence of functions, has been corrected. Problem occurred when some time domain functions left files open, and other functions tried to open these files, and "access denied" message would appear.

16) Correction to Ariel DSP-C40 driver bug, where setting first value in input lists or first trace to channel 1 did not work.

Release Notes, Hypersignal-Macro Series, MAR94-JUN94

Release notes for Hypersignal-Macro series software from March 1994 through June 1994 cover Hypersignal-Macro RT and Hypersignal-Macro EX v4.00xa - v4.00xc, Hypersignal-Acoustic

v4.00xa - v4.00xc, and Hypersignal-Macro NRT v5.00xa - v5.00xc.

New Features, Hypersignal-Macro Series, MAR94-JUN94

1) Selection of trace indicators inside Spectrum Analyzer now update x-axis units and labels. Also, for mixed linear and log frequency x-axis display (during dual-trace operation), separate log and linear grids are used.

2) Impulse response attribute option added to File Acquisition function.

3) First version of support added for Signalogic Sig32C-8 8-channel board.

Enter SIG32C in System Configuration menu. Instrument operation allows any two channels to be selected; continuous disk acquire/generate allows any combination and number up to the maximum eight channels. See the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for details on using this board and data-entry menu usage.

4) Time domain waveform display function speed increase when only one trace is being displayed.

5) A "string" filename can be entered in the Difference Equations function in place of an equation, causing Hypersignal to read the specified file as ASCII and retrieve an equation. The equation in the

file can be up to 255 characters long (the total menu limit in Hypersignal is 80). Also, the equation can contain spaces. The equation should be entered on one line ONLY in the file. The filename should be entered in the Difference Equation equation field as

```
xxxxx.str
```

where xxxxx is the filename. The .str extension is REQUIRED. Any drive and subdirectory information entered in the Difference Equations equation fields will override the default User Setup parameters.

6) /PD:\path\ command line switch added that allows the waveform drive and path information to initialized in the Hypersignal command line, where D is a drive identifier, and path is subdirectory information.

7) New macro language functions:

; returns 1 if last function was "escaped", or aborted, returns 0 if last function

was
execut
ed

;
sends
messa
ge to
extern
al
device
driver
askin
g for
updat
ed
functi
on/dis
play
param
eters;
used
just
prior
to
$RU
N or
$RU
NINT
ERA
CTIV
E
comm
ands

;
causes
displa
y to
refres
h
witho
ut

changing any screen or hardware settings; has an equivalent ctrl-L keyboard interactive command;
causes DSP board to reset without changing any screen or hardware settings; has an equivalent ctrl-Enter keyboard

interactive command

; causes macro language compiler to create a text file, and write the string data to it (like the {INC directive, no executable code as a result of the "{ }" line itself is produced)

8) Gaussian window option added to FFT Generation function.

9) Minimum phase (minimum delay sequence) calculation macro function added to Hypersignal-

Macro EX and Hypersignal-Acoustic distribution diskette #2. For more information on min phase function, see the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2).

10) Signal generation looping has been enhanced to allow easy construction of a continuous function generator. By creating short signal/function waveforms using Difference Equations, and then using these waveforms as input to the Analog Conversion function, a continuous signal generator can be achieved when the ACQ/GEN field is set to 'GL'. See the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for guidelines on how to perform continuous function generation on one or more channels.

11) Signal acquisition drivers for some DSP boards enhanced to accommodate the possibility of multiple channel operation, including odd numbers of channels. A direct result of work performed for Signalogic Sig32C-8 support (see above).

12) Addition of "constant bandwidth" frame command in Magnitude and Phase Display functions. Using the spacebar to pan forward (or the backspace key to pan backwards) causes the displayed bandwidth to remain the same in the next (or previous) frame. Using the tab key to pan forward (or the Shift-tab key to pan

backwards) causes the displayed bandwidth to be incremented (or decremented) within the current frame.

13) Addition of $Y,INTERACTIVE command to $ macro language commands. This command is intended for use with the Analog Conversion function; it answers a prompt with 'Y' (yes) and causes the function to enter interactive mode at the same time. This allows disk record/playback operations to still run to completion as specified in a macro language file, yet remain sensitive to keyboard input during operation.

14) Addition of GetExePath() function to macro language. Assigning the result of this function to a string variable returns the drive and subdirectory (path) from which Hypersignal was executed.

15) Addition of "Save", "Save As..", and "New" commands to the Waveform Display/Edit function. Press the F5 key inside the display to save (update) a waveform currently being edited, press the F6 key to save the waveform as a new filename (and change the display for the current trace to the new waveform), and press the F7 key to create a new wave- form that inherits the attributes of the current waveform, but zeros all sample points. Commands apply to the waveform displayed in the currently selected trace (see '1' and '2' commands).

Bug-Fixes/Corrections, Hypersignal-Macro Series, MAR94-JUN94

1) Correction to macro language compilation where string constants (text enclosed by "" marks) that contained macro language keywords ('end', 'procedure', etc.) could be misinterpreted by the compiler.

2) Lowercase characters allow in string constants.

3) Correction to PutDataValue function which incorrectly altered the Maximum Amplitude value in the destination waveform file during some PutDataValue calls.

4) Macro language run-time errors in Users Manual Appendix A should be listed at starting at 100 instead of 70.

5) Problem corrected where repeated DSP/Analog board function usage would eventually result in "....DSP PROGRAM FILE NOT FOUND". Problem caused by temporary file not being closed in all cases.

6) Problem with DSP3210-based stimulus & response corrected (affected Ariel DSP-3210 board and CAC QUANTUMdsp board).

7) Problem with instrumentation real-time filtering applied to second trace in with DSP32C and DSP3210 boards corrected.

8) Correction to problem with 2-D Spectrograph display where very

low amplitude waveforms did not display correctly.

9) Correction to File Acquisition problem in macro language mode, where presence of existing output file could cause macro to hang.

10) Correction to external clock problem with Data Translation DT282x series boards.

11) Correction to 10% sampling rate error problem with CAC QUANTUMdsp DSP3210 boards.

12) Addition to Ariel DSP-96 driver so that both "pre-June 1993" memory-mapping PALs and post-June 1993 PALs are recognized, and Hypersignal buffer locations automatically adjusted. Also, a software update was made to allow the 80x86 INSTRUCTION SET to be enabled in faster machines for much better overall performance without a loss in reliability. One negative result of this change: real-time filtering coefficients and delay storage elements for ALL DSP96002 boards had to be relocated into onchip X: and Y: RAM at $40 thru $1EF; otherwise, the higher transfer rates caused a conflict with simultaneous bank DMA accesses in some situations. Moving filter data to internal RAM avoids the problem and allows maximum DMA and host-port throughput. If you need longer filters, disable the 80x86 INSTRUCTION SET field and talk to Signalogic about making a slight modification to the DSP96002 source code (i.e. using the DSP

Source Code Interface product for the DSP96002).

13) Correction to TMS320C25 iFFT function, which somehow became broken and was incorrectly calculating maximum amplitude of output data.

14) Increase in TMS320C30 and TMS320C40 maximum FFT size to 4096.

15) Repair to external program execution, which in some cases didn't use explicit path entry in the Ext Prog Config menu selection.

16) Addition of Oak Technology video drivers to Graphics Config menu. These drivers were inadvertently left out, and have been added to the end of this menu. You must use v3.7C of the MetaWindow TSR for these drivers to work. v3.7C corresponds to a June 6, 1991 date on the met_shel.exe file (distribution diskette #2).

17) Addition of "active clear" code for some video boards that do not erase background properly when used under Windows 800x600 and 1024x768 video drivers. Only boards known to have this problem have been affected.

18) Correction to execution of batch files from User-Defined menu items (which are set up in Ext Prog Config menu selection). In some cases, batch files would not execute because command.com was not searched for in the system

path. This problem has been corrected.

19) Correction to "incomplete grid lines" plot in Waveform Display/Edit when framesize exceeds 500,000 samples.

20) Correction to problem with waveform acquisition (Analog Conversion function) where waveform file lengths in excess of 4 Msample could fail on first attempt.

21) Correction to D/A playback sampling rate selection problem from Waveform Display/Edit where two display waveforms had different numbers of channels.

22) Correction to uninitialized variable in 3-D Spectrogram display, which infrequently caused log mode (amplitude) operation to display incorrectly.

23) Correction to multichannel file stripping in Analog Conversion input, where the base filename (for interleaved waveform file) was truncated to six characters instead of seven in cases where less than 10 channels were acquired.

24) Correction to dual-channel Waveform Display/Edit display when the second channel display could possibly show an artifact at buffer boundaries (at 8192 points or 16384 points).

25) Correction to time axis in Waveform Display/Edit function when two dual-channels waveforms (4 channels total) are displayed at the same time. The

axis was incorrect by a factor of two.

Release Notes, Hypersignal-Macro Series, AUG93-FEB94

Release notes for Hypersignal-Macro series software from August 1993 through February 1994 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.40i, Hypersignal-Acoustic v3.40i, and Hypersignal-Macro NRT v4.40i.

New Features, Hypersignal-Macro Series, AUG93-FEB94

1) A new version of compiler has allowed reduced memory usage by HS-Macro and HS-Acoustic. Much more efficient usage of overlays, especially where hardware device drivers are concerned, has been achieved. Up to an additional 50 kb may be available at times for running external user-defined programs. NOTE: Hypersignal currently supports over 55 different types of DSP/Analog hardware. Most, but not all, of these hardware types are currently available for testing at Signalogic. If any DSP/Analog hardware problems should occur with version 3.40i or later, please contact Signalogic at once; we will obtain the necessary version and configuration of hardware to match yours, and get a working version of software to you as soon as possible.

2) First version of Ariel DSP-C40 board (TMS320C40) support installed. Enter 'ARIC40' in System Config menu.

3) New macro language additions and improvements have been installed, including:

New Commands and Operators

; sets screen dimensions of following waveform display; use before $RUN

; sets screen offset of following waveform display; use before $RUN

; sets handle of target window when sending information to Windows application

;
modulus operator, performs modulus division operands and type conversion rules are the same as integer division operator (div)

General Improvements

- spaces can be included in macro language file parameters; the HS-Macro
- editor will strip them out (for example, string values of " ", or "" can be used to blank a menu field)
- macro language source lines can now be up to 255 characters long
- up to 16 parameters can now be passed to procedures

4) A '/d' "driver mode" flag can be invoked on the command line which will cause HS-Macro to invoke no graphics displays, and make no keyboard inquiries--avoiding user I/O hardware accesses altogether. This flag should be specified only when HS-Macro has been remotely invoked from an external DOS or Windows application (using a device driver to talk to

Hypersignal), and has been instructed to run a macro language file at startup. The macro language file should contain no displays that use the $MANUAL or $WAITFORKEY commands. Below is an example of a WinExec call to do this:

```
StatusCode =
WinExec("C:\\HSIGNAL\\H
SMACRO.PIF /d
fname.mac", SW_HIDE);
```

A device driver such as pipe.386 (included with DSPower) should be installed in the system prior to using this mode. The HSMACRO.PIF file should be set up as a windowed application, 384 kb required, and background operation; I/O ports and memory need not be monitored. No

memory needs to be locked. Be sure that fname.mac includes an Abort or Exit command in order to terminate the process. Also, DO NOT invoke the HSM.BAT file, as this will install the MetaWINDOW graphics driver, which is not needed.

5) An 'EXPORT HEADER (Y/N)' field has been added to the ASCII Export function. Enter a 'N' to cause Hypersignal-Macro waveform headers to not be included in the exported ASCII file. NOTE: If you desire to re-import a waveform file exported without its header, DO NOT use the ASCII Import function. Instead, use the File Acquisition function and set the SAMPLES/BYTES TO DISCARD field to zero.

Bug-Fixes/Corrections, Hypersignal-Macro Series, AUG93-FEB94

1) Correction to paste-mode Delete function in Waveform Display/Edit, where delete operations could cause bogus waveform data to appear at the end of the waveform.

2) Correction to horizontal axis readout in Waveform Display/Edit function, where the righthand value, when displayed in time units (not sample units), could be the equivalent of one sample too large.

3) Correction to Transfer Function Generation during macro operation, which would crash if the output waveform files already existed.

4) Further correction to Transfer Function Generation where large amplitude results in output could cause crash.

5) Correction to uninitialized variable problem for Intelligent Instrumentation PCI20202C system where modules could be incorrectly identified on carrier board.

6) Correction to string parameter passing in macro language procedures where parentheses and commas were being stripped from string values; e.g. a string such as "sin(2*pi*n)" can now be passed as a procedure value without problem.

7) The macro record mode has been modified to insert a $RUN command after a $PRINT command, which correctly allows a recorded print command to run correctly when "replaying" the macro.

8) Invoking the ANALOG CONVERSION menu with no DSP/Analog hardware selected in macro mode now produces a macro error message and proper abort sequence.

Release Notes, Hypersignal-Macro Series, JUN93-JUL93

Release notes for Hypersignal-Macro series software from June 1993 through July 1993 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.40h, Hypersignal-Acoustic v3.40h, and Hypersignal-Macro NRT v4.40h.

New Features, Hypersignal-Macro Series, JUN93-JUL93

1) First version of MTT Instrumentation DSP4200 system (TMS320C30) support installed. Enter 'MTTC30' in System Config menu. Hypersignal-Nihon only.

2) First version of CAC QUANTUMdsp DSP3210 multimedia device driver installed. Enter 'CAC3210' in System Config menu. NOTE: the QUANTUM.EXE file must be run one time beforehand. Placing 'quantum -q' into your autoexec.bat file is a good idea (this program initializes the FPGAs on the QUANTUMdsp board, and is NOT a memory-resident program). The QUANTUM.EXE, VCOS.CFG, and QUANTUM.V4 files must be available (VCOS.CFG and QUANTUM.V4 are used as input by QUANTUM.EXE). These files are installed from the OEM diskette provided by CAC. If there is any problem in locating or using these files, contact Signalogic.

3) First version of Turtle Beach MultiSound DSP56001 device driver installed. Enter 'TBS56' in System Config menu. NOTE1: the TBS56 driver defaults to a "DSP56TBS.OUT" file, instead of the standard "DSP56001.OUT" file used for all other supported DSP5600x boards. NOTE2: The MultiSound board allows only about 12k X 24 X: and 12k X 24 Y: of data memory, which somewhat

limits Hypersignal functions such as Spectrum Analyzer, DSP-based convolution, and stimulus & response.

4) First version of National Instruments AT-DSP2200 (DSP32C) device driver installed. Enter 'NAT32C' in System Config menu.

5) "Stimulus flag" installed in Analog Conversion function. Entering ',S' after stimulus filename will cause stimulus not to repeat, regardless of response (record) length. After reaching the end of the stimulus waveform, zero data is output to the hardware D/A converters. Entering ',R' after stimulus filename will cause stimulus to repeat (as will blank, or default, entry).

6) All Hypersignal programs now require Motorola DSP5600x ".OUT" (COFF format) files instead of ".LOD" files. This brings Hypersignal current with Motorola 5.0 tools.

7) Activation of dual (independent) channel filtering in Analog Conversion function. Enter 'fname1/fname2' in place of 'fname' in CHANNEL LIST field. See the Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for more information.

8) Improvements made to PutDataValue function in macro language, including allowing the function to expand an existing waveform file by writing beyond the end of the file. See the

Hypersignal-Macro Series software addendum.doc file (distribution diskette #2) for more information.

9) SetFileStripMode(N) and $SETFILESTRIPMODE(N) interactive commands added to macro language. Invoking these commands (for example, invoking $SETFILESTRIPMODE(ON) inside Analog Conversion construct) will cause waveform file-stripping to occur (for multichannel waveform acquisition) if N is set to 'ON' (or '1'). N set to 'OFF' or '0' disables file-stripping (default mode).

Bug-Fixes/Corrections, Hypersignal-Macro Series, JUN93-JUL93

1) Correction to page 2 video problem in instrument display in HS-Nihon.

2) Correction to errant Kanji character problem in HS-Nihon.

3) Correction to Ariel DatPort driver with PC-56D. Both ProPort and DatPort DSP code interrupt service routines were changed to accommodate subtle but significant differences between these two units.

4) Correction to Texas Instruments EVM TMS320C30 device driver.

5) Correction to '<' comparisons in macro language, where a run-time error or bogus results could occur. '>', '<=', '>=' comparisons did not exhibit this problem.

6) Correction to clipping problem past 4096 samples of output in Difference Equations function, which occurred in some cases where input waveforms were involved, and output amplitude differed substantially from input amplitude.

Release Notes, Hypersignal-Macro Series, APR93-MAY93

Release notes for Hypersignal-Macro series software from April 1993 through May 1993 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.40g, Hypersignal-Acoustic v3.40g, and Hypersignal-Macro NRT v4.40g.

New Features, Hypersignal-Macro Series, APR93-MAY93

1) Spectrum Analyzer displays horizontal axis in time units for raw analog time input or impulse response display if:

- the only trace active is one of these, and
- two traces are active and bottom trace is one of these

2) For Hypersignal-Acoustic only, the ctrl-U key can be used to cause distance units readout (based on the speed of sound) instead of time units readout in all displays where appropriate.

3) Using 'ctrl-Enter' instead of 'Enter' in the Spectrum Analyzer resets only the DSP, and not current display parameters. This

can be useful for trigger reset or averaging reset.

4) Significant improvements to sampling rate conversion macro (SAMCONV.MAC), including using zero-sample insertion to avoid sin(x)/x effects and calculation of LPF transition band and cutoff frequency as a percentage of desired frequency shift.

5) Major speed improvement to all real-time functions using an AT&T DSP3210-based board.

Bug-Fixes/Corrections, Hypersignal-Macro Series, APR93-MAY93

1) A long-lived and troublesome bug in Waveform Display/Edit, where changes made in point mode did not always "take" after a mode switch, escape to edit menu, zoom, or frame translate has been squashed.

2) Correction to cursor amplitude readout in Spectrum Analyzer during transfer function display after engineering units have been set.

3) Correction to horizontal axis readout in Waveform Display/Edit function for time units. One sample less than correct value was being used to display righthand axis value.

4) Correction to memory requirement determination for TMS320C30, DSP32C, DSP3210, and DSP96002 boards for

Convolution/Correlation with very long impulse response (greater than 12,000 points). Previously, in some situations, inadequate amounts of memory were not observed.

Release Notes, Hypersignal-Macro Series, MAR93

Release notes for Hypersignal-Macro series software in March 1993 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.40f, Hypersignal-Acoustic v3.40f, and Hypersignal-Macro NRT v4.40f.

New Features, Hypersignal-Macro Series, MAR93

1) Updates to Hardware Reference Guide (chapter 4) and Users Manual (chapter 7). Current dates are March 1993. Tutorial and Software Reference Guide are unchanged (current dates are January 1993, and August 1992, respectively).

2) Addition of GetSysConfig, StrPos, and UpCase functions to macro language. The GetSysConfig function allows macros to get and store System Configuration menu fields. The StrPos function searches one string variable for the first occurrence of another. The UpCase function converts a string to all uppercase characters. See Users Manual section 7.10 for fully detailed information.

3) Addition of WriteDSPMem and WriteDSPVal functions to macro language. These functions allow macros to access the currently installed DSP board, and allow user-defined DSP code access to internal Hypersignal values and menu values normally communicated to DSP/Analog boards. See also new ClearDSPMem and ClearDSPVal functions. See Users Manual section 7.10 for fully detailed information.

4) Macro language string constants have been updated to allow spaces.

Bug-Fixes/Corrections, Hypersignal-Macro Series, MAR93

1) Correction to display in Spectrum Analyzer of transfer function waveform files previously stored by the Spectrum Analyzer (for DSP56001, TMS320C25, and ADSP-2101 based boards).

2) Correction to macro language compiler where assignment statements of form a = -2 + b, or a = +3 - b caused an error.

3) Correction to macro language compiler where the "EXPONENT" constant in PutHeaderValue and GetHeaderValue function calls did not work correctly.

4) Correction to Recursive Filtering and Difference Equations functions for input waveforms with a stored waveform header framesize greater than 32767.

5) Correction to Power Spectra Generation when invoked from a macro language file.

Release Notes, Hypersignal-Macro Series, FEB93

Release notes for Hypersignal-Macro series software in February 1993 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.40e, Hypersignal-Acoustic v3.40e, and Hypersignal-Macro NRT v4.40e.

New Features, Hypersignal-Macro Series, FEB93

1) Updates to Hardware Reference Guide and Tutorial. Current dates are January 1993. Users Manual and Software Reference Guide are unchanged (current dates are August 1992).

2) Executable DSP filenames can be indicated in the System Configuration menu. Enter the filename after the board designator in any or all of the first four fields. Examples:

```
ATT32C,dsp32c.out

ZPB34-
zpd1007,new32c.out

ARI56,dsp56.lod

LSI25,tms25.out
```

Bug-Fixes/Corrections, Hypersignal-Macro Series, FEB93

1) Work-around in device driver for LSI DSP56001 boards manufactured in UK (i.e. not Spectrum-manufactured version) where second channel in two-channel acquisition or instrumentation operations would sometimes show duplicated samples. The new device driver works with both types of boards.

2) Several enhancements to SYSid file read utility (SYSIDRD.EXE). Can now be run from a macro (SYSIDRD.MAC), accepts both binary and ASCII formats, correctly reads frequency responses with magnitude greater than 1.0, and has more flexible user-interface and file specifications. Hypersignal-Acoustic only.

Release Notes, Hypersignal-Macro Series, DEC92-JAN93

Release notes for Hypersignal-Macro series software from December 1992 through January 1993 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.40d, Hypersignal-Acoustic v3.40d, and Hypersignal-Macro NRT v4.40d.

New Features, Hypersignal-Macro Series, DEC92-JAN93

1) Addition of Intelligent Instrumentation ZPD 1007 analog I/O option for basic ZPB34 board selection in System Configuration menu.

2) Addition of MTT Lory DSP4100 system operation. Enter 'MTT32C' in System configuration menu (active in Hypersignal-Nihon only).

Bug-Fixes/Corrections, Hypersignal-Macro Series, DEC92-JAN93

1) Correction to stimulus & response operation when the "monitor mode" has been enabled; problem was manifested as stimulus not ending or repeating properly, or same as when monitor mode was not enabled (which operated correctly).

2) Correction to 2-channel continuous disk record for AT&T DSP3210 boards.

3) Experimental correction to Ariel DSP-96 FFT Generation and iFFT Generation hang-up problem.

4) Correction to DSP96002 iFFT results for all DSP96002 boards.

Release Notes, Hypersignal-Macro Series, OCT92-NOV92

Release notes for Hypersignal-Macro series software from

October 1992 through November 1992 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.40a - v3.40c, Hypersignal-Acoustic v3.40a - v3.40c, and Hypersignal-Macro NRT v4.40a - v4.40c.

New Features, Hypersignal-Macro Series, OCT92-NOV92

1) Addition of Spectrum Analyzer improvements listed in 10 SEP 92 versions (see below) for DSP56001-based boards and DSP96002-based boards.

2) Addition of PC-56D ProPort and DatPort driver (Hypersignal-Acoustic only). Enter 'ARI56-D' in System Configuration menu.

3) Macro and external utility to allow time-reversed 3-D Spectrogram viewing (to enable viewing of features in the data "hidden" by high amplitudes that occurred early in time). Enter "freverse" in Macro Manager menu. Hypersignal-Macro EX and Hypersignal-Acoustic only.

4) Macro and external utility to read CADP2 program ETC output ASCII files (CADP2 is modeling/simulation program created by JBL). The macro produces a Hypersignal impulse response time domain waveform. Enter "cadp2rd" in the Macro Manager menu. Hypersignal-Acoustic only.

5) Addition of Ariel DSP-3210 board. Enter 'ARI3210' in System Configuration menu.

6) Further increase in speed of loading for DSP32C-based board COFF files.

7) Increase in speed of continuous transfer from disk to D/A output for TMS320C30-based boards.

8) Addition of Peralex PROTO-56 board. Enter 'PA56' in System Configuration menu.

Bug-Fixes/Corrections, Hypersignal-Macro Series, OCT92-NOV92

1) Correction to Difference Equations function when referring to long x1 and x2 waveforms that contain waveform header exponent<>0.

2) Correction to GetHeaderValue function in macro language when reading waveform header amplitude and framesize values, which are word. Function was reading them as integer, which caused values above 32767 to be read incorrectly.

3) Correction to problem introduced in previous version with pan function in Magnitude and Phase displays (log frequency axis mode only).

4) Correction to real-time filtering when using analog output of TMS320C30-based boards.

Release Notes, Hypersignal-Macro Series, SEP92

Release notes for Hypersignal-Macro series software in September 1992 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.30d - v3.30e, Hypersignal-Acoustic v3.30d - v3.30e, and Hypersignal-Macro NRT v4.30d - v4.30e.

New Features, Hypersignal-Macro Series, SEP92

1) Addition of "start frequency" location in extended waveform header that allows accurate frequency-axis labeling of frequency-zoom results that do not begin at zero frequency. Currently, only the TEFRD utility (Hypersignal- Acoustic only) writes this location, although it is expected other functions will use it in the future (such as the FZOOM macro).

2) Frequency domain displays enabled to display the new "start frequency" value in extended waveform header (if not zero) correctly for trace 1 waveforms.

3) Further enhancements to TEFRD utility (Hypersignal-Acoustic only), including ability to abort operation, better error-checking to discover files not in correct SoundLab format, and correction to operation with full drive and subdirectory specification for the input SoundLab waveform file.

4) New revision of documentation:

- Rev. 3.2 of Users Manual
- Rev. 3.1 of Hardware Reference Guide
- Rev. 2.1 of Software Reference Guide
- Rev. 2.1 of Tutorial

All documentation should have "August 1992" date on front page. All Hypersignal software purchased from Signalogic is now delivered in a three-binder set. The first binder contains the Users Manual, the second the Hardware Reference Guide and Software Reference Guide (separated by a tab divider) and the third the Tutorial.

5) Waveform header exponent increased to 8-bits, and stored in new file header location (see Signalogic DSP Software Reference Guide section 4.1, Hypersignal Waveform File Format), increasing the dynamic range of data that can be represented by Hypersignal waveforms to the following range: 2.93873e-39 to 5.575e+42.

6) Several Spectrum Analyzer improvements:

- Addition of new data type: "impulse response". This option enables an inverse FFT of transfer function output, allowing you to see the time domain impulse response of a system under test. Averaging can be applied. Enter 'I' or 'Inn' in the TRACE1 or TRACE2 fields, where "nn" is the number of frames to average. In this version (3.30d), *this function is currently working only for boards with DSP32C and TMS320C3x DSP chips.*

- Averaging can be invoked for any type of data except for time domain; for example, to see an averaged phase trace, enter 'PHnn', where nn is the number of frames to average. Note: currently, a 'Mnn' or 'Mnn+' entry displays the same data as a 'Pnn' or 'Pnn+' entry.
- The real and imaginary output from a transfer function trace display can be written to disk file using the WRITE command. Answer the prompt to create .IMA and .REA files suitable for inverse FFT processing. This option is in Hypersignal-Acoustic only. NOTE: the real and imaginary data cannot be subjected to averaging. In this version (3.30d), this function is currently installed only for TMS320C3x and DSP32C boards.
- A file input can be simultaneously shown with transfer function display; enter the filename in the TRACE1 field, and specify transfer function display in the FORMAT2 field. The system assumes that the "other" channel than what is shown in the TRACE2 field should be used as the numerator channel.
- Transfer function display (and its new relative, impulse response display), can be invoked on same-channel setups, where the only difference could be real-time filtering, for example.
- Any FORMAT1 or FORMAT2 entry specifying averaging of linear or log magnitude data ("nn" frames entry after format) and referring to a TRACE1 or TRACE2 entry specifying a waveform file will automatically retrieve the power spectra file (averaged) version of the waveform. A format entry of 'P' (for power spectra) can still be entered, but does not need to be if averaging is entered.
- Single-trace display during dual-trace operation can more easily be specified by either entering '1' or '2' in the FORMAT subfield of the TPL field, or blanking the F1 or F2 field (or using an 'N' entry), depending on which trace should be disabled. Also, this type of entry now disables extra labeling for the trace not being displayed.

7) Stimulus output in Analog Conversion changed so that stimulus repeats immediately after end of stimulus waveform is reached. To deactivate repeat, or increase interval between repeats,

the stimulus waveform must be created with as much zero data as necessary appended to end of actual stimulus data.

8) Frequency zoom macro (fzoom.mac) added to diskette #2. An arbitrary segment of bandwidth can be selected for high resolution frequency analysis.

9) Stimulus & response macro (stim.mac) added to diskette #2. A linear sweep sine wave is created and used as a stimulus; the response is cross-correlated with the sweep to extract an impulse response; the macro ends with the frequency response displayed and display controls in manual mode. The sweep length can be chosen (up to 16384 samples) to control frequency resolution.

10) A RESTORE DOS PATH field added to the External Program Config function that allows the system to automatically restore DOS path settings altered by external programs or by users in a DOS shell. The field can be disabled to provide operation as with previous versions.

11) Size of Directory Display data-entry field increased.

12) Power Spectra Generation FRAMES TO AVERAGE field is now interpreted as a percentage when straight averaging is used ('A' entry in METHOD field). Its interpretation with other methods remains unaffected.

13) Sampling rate conversion macro (samconv.mac) speed of execution improved by replacing Parks McClellan design method in interpolation/anti-aliasing filter with Kaiser Window method. Also, the final difference equation in the macro has been modified to remove the significant delay inserted by the longer filters typically produced by the Kaiser Window method.

14) "DOS Shell" command added to macro language. See the Hypersignal-Macro Software series addendum.doc file (distribution diskette #2) for more information.

15) A fourth mode of log amplitude display has been installed in frequency domain file-based displays (Magnitude Display, 3-D Spectrogram, 2-D Spectrograph) that refers to the system value for 0 dB, but does not use the Spectrum Analyzer (S.A.) calibration offset. The modes are now as follows:

mode/description

autoscale; 0 dB is maximum |amplitude| in waveform

0 dB is system reference; S.A. calibration offset not added

0 dB is system reference; S.A. calibration offset added

0 dB is system reference; WH calibration offset added ("WH" is waveform header)

Bug-Fixes/Corrections, Hypersignal-Macro Series, SEP92

1) Correction to directory search on power spectra waveforms in frequency domain displays.

2) Correction to waveform file write function ('W' key) in Waveform Display/Edit display that was not correctly transferring waveform exponent from displayed waveform to new output file waveform, causing amplitude of new output file waveform to be incorrect.

3) Several menu areas corrected that didn't display proper color values when a monochromatic graphics selection had been made from the Graphics/Mouse Config menu.

4) Continuous recording to disk for DSP Research TIGER 30 (with TLC32044 analog I/O option only) updated to discard first A/D points, which contain "circuit warmup" and settling artifacts.

5) Correction to problem in all TMS320C30 board-based real-time filtering in analog I/O-related functions, where repeated "re-starts" of the function (starting from the data-entry menu) would cause the filtering to be intermittent (not always initialized correctly)

6) Correction to type checking in macro language compiler where cases of

int = int / int

were not detected as assignment mismatches, allowing "/" (real division) results to be incorrectly assigned to integer output.

7) The Analog Conversion function has been corrected in macro mode to not return to the data-entry menu

after a $RUN command. After the function has completed, more interactive commands can now be processed, and the function can be run again if needed. Use a '$ESC' command to return to the main menu as appropriate.

8) Correction to Correlation function when both input waveform and filter waveform were equal to the maximum filter length (16384).

9) The $INS and $DEL interactive commands now correctly perform waveform insert and delete operations in the Waveform Display/Edit function from a macro. See the "CUT.MAC" example file.

10) The D/A output command now correctly works when invoked from a macro file (using '$D' interactive command).

11) Correction to time data display in overlaid-trace mode in the Spectrum Analyzer, where the trace was not centered about zero in the display area.

12) Setting dual- or overlaid-trace controls to "lock-step" mode in the instrumentation displays now correctly sets both "active trace" indicators.

13) Convolution/Correlation activated for LSI ADSP-2101 boards that have memory upgrade installed (8 kW prog, 12 kW data).

Release Notes, Hypersignal-Macro Series, JUL92-AUG92

Release notes for Hypersignal-Macro series software from July 1992 through August 1992 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.30a - v3.30c, Hypersignal-Acoustic v3.30a - v3.30c, and Hypersignal-Macro NRT v4.30a - v4.30c.

New Features, Hypersignal-Macro Series, JUL92-AUG92

1) Support for the Sonitech Spirit-30 / SAIB (Stereo Analog Interface Box) added. Enter 'SONC30-SAIB' in the System Configuration menu. Hypersignal- Macro supports one- or two-channel operation, line-level input/output, and programmable sampling rates (ranging from 5 to 48 kHz) from standard menus. Although the SAIB supports many additional, powerful functions, including

- programmable selection of input source (line-level, microphone, handset, telephone line)
- programmable data format (8-bit uLaw or ALaw)
- programmable hardware-controlled input gain and output attenuation
- programmable headphone enable/disable, and
- programmable telephone line control and status functions,

these functions currently cannot be controlled from Hypersignal-Macro menus, but must currently be set using two control words stored in

the hsmacro.cnf and hsacous.cnf configuration files. See the Hypersignal-Macro Series software addendum.doc file and config.doc files (distribution diskette #2) for more information. NOTE: Hypersignal-Macro expects the SAIB to be connected to serial port 1 on the Spirit-30 (bottom connector of backplane), allowing Sonitech IC-100 type analog I/O units to be connected to serial port 0.

2) Setup "save" files incorporated into the File Acquisition function, to facilitate re-performing file import of older files and/or files requiring non-standard settings. Menu setup save files are ASCII; the File Acquisition filename extension is '.ACQ'.

3) Maximum value for the framesize setting in File Acquisition and Analog Conversion functions (used only in subsequent post-processing with other functions) increased to 65535.

4) The Display function in the macro language has been modified to begin display in the first line of the System Status area and then use successive lines (up to 6) for display. The next Display command will clear the area and start over at line 1 again. At some future point, it is tentatively planned that scrolling and "ClearDisplay", "GotoDisplay", etc. commands (or similar) may be installed in the macro language to

facilitate control over the display area.

5) Amplitude (vertical) scaling in the Waveform Edit/Display function when in log amplitude display mode has been changed to conform with Hypersignal- Macro frequency domain scaling. PgUp scaling (zoom in) now expands the display with the top fixed (instead of the bottom). To see different vertical sections of the waveform display, set the LG subfield value of the AMP SCALING field accordingly.

6) Support for "revision B" firmware on the Motorola ADS56000 development system board. System Configuration menu entry for the ADS56000 system and its analog I/O options remains the same, as Hypersignal-Macro determines the revision level of the firmware automatically.

7) Maximum amount of the impulse response (filter) waveform utilized in the Convolution/Correlation function increased from 8192 to 16384 samples. Stereo impulse response maximum increased from 4096 to 8192.

8) Addition of "LN" and "LG" subfields to the time domain Waveform Display/Edit AMP SCALING field. Allowing entry of both linear and log values eliminates the need to change this field every time the time domain amplitude display mode is changed.

9) Addition of "amount per color" read-out in 2-D Spectrograph display in linear display mode. Previously, this read-out appeared only in log display mode.

10) New, external "TEFRD" utility that reads TEF20HI analyzer TDS (frequency domain) and ETC (time domain) files into Hypersignal format. Use either the TEFRD.EXE utility directly or the TEFRD macro from inside Hypersignal (Hypersignal-Acoustic only). NOTE: the ETC output of this utility is NOT adjusted for the sampling rate conversion performed by the TEF20HI, and does NOT necessarily depict reflections and other physical phenomenon accurately.

11) Macro Language Improvements:

- Input parameters (#1 through #6) can now be passed in the HSM batch file command line, after a macro filename. For example, invoking the command HSM fname.mac #27 #tempest from the DOS command line, a batch file, or from another program will invoke Hypersignal-Macro (or other version, depending on the batch filename), cause the macro file "fname" to compile/run immediately, and pass the strings "27" and "tempest" to the first two input parameter fields in the macro manager menu. The "#" symbol MUST precede the actual parameter value.
- External programs (user-defined programs) can now be invoked from macro language files by either their current Ext Program Config menu labels, or the following generic commands: User-Defined Time 1, User-Defined Time 2, User-Defined Freq 1, User-Defined Freq 2, User-Defined Util 1, or User-Defined Util 2. The generic commands were added because in some cases the Ext Program Config menu needed to be configured from the macro file, at run-time; thus, program names might not be known to the program at compile

time. Also: any menu parameters entered after a User-Defined program invocation are now treated as command-line values passed to the external program. Of course, this only works if the external program accepts command-line values. Up to 10 menu parameters can be entered after a User-Defined program invocation. Remember to always include an "endfunc" statement after any menu command in a macro file.

- PutHeaderValue(filename, position, value) and PutDataValue(filename, position, value) functions have been installed. Filename should specify a string variable or constant. Position should be specified in words--see Signalogic DSP Software Reference Guide section 4.1, Hypersignal Waveform File Format. Value should be an integer or real string variable or constant.

- var = GetUserSetup(N) function has been installed. Var should be a string variable. N should specify the User Setup area parameter to be returned, ranging from 0 to 7. GetUserSetup always returns string values.

12) New, external "FORMANT" utility that extracts formants from frequency domain waveform files (presumably speech files) and creates a new magnitude file as output showing the extraction.

13) '-NN' clock rate entry is now possible following DSP/Acquisition board entry in the System Configuration menu. NN should be a value in MHz, and is intended for boards which

- base their analog sampling rate on the main DSP clock, typically using an integer division, or submultiple, method

- are available in more than one clock rate version

For example, the Intelligent Instrumentation PCI20202C is a TMS320C25-based board and comes in three speeds: 28 MHz, 40 MHz, and 50 MHz. Default entry of 'PCI202C' would specify 28 MHz operation (so would

'PCI202C-28'). Entry of 'PCI202C-50' would specify 50 MHz operation. As another example, the DSP Research Tiger-30 board comes in 33.21 and 40 MHz versions, and the AIC analog I/O option uses these rates divided by 4 as a master input clock. Blank (default) entry is highly recommended unless you are absolutely sure you have a non-standard clock rate board.

Bug-Fixes/Corrections, Hypersignal-Macro Series, JUL92-AUG92

1) A check has been installed to prevent AMP SCALING field entry in frequency domain displays to be set such that top and bottom amplitude values can be equal (except for zero values, which, as before, are the default entry and are interpreted as autoscaling).

2) Correction to stimulus & response function for all TMS320C30-based boards where a filter filename entered in the Analog Conversion menu did not actually enable filtering.

3) Correction to storage of amplitude value in waveform headers by Analog Conversion and File Acquisition of integer data with minimum negative value of -32768.

4) Repair to the non-interactive version of the WaitForKey function in the macro language.

5) Correction to accelerated functions invoked from macro language files for the Sonitech Spirit-30 board. A reset timing problem caused this board to produce invalid results in macro mode (accelerated functions only).

6) Correction to editing in the Waveform Edit/Display function when in log amplitude display mode.

7) Correction to accelerated Cross Correlation function (all DSP boards) problem where starting point was incorrect (extra zeros were inadvertently attached to the beginning of the input waveform).

8) Correction to Parks-McClellan design where recent improvements had introduced a problem preventing design of all-pass delay filters.

9) Correction to results of the Convolution/Correlation and Recursive

Filtering functions when intermediate overflow occurred. In such cases, the waveform exponents were being calculated correctly, but not added to a previously existing waveform exponent, if any. Also, the scaling prompts (asking for permission to scale) have been removed from these functions, as they are now redundant due to the new waveform file header exponent system.

10) Correction to transfer function operation for all DSP56001-based

and TMS320C25-based DSP/Acquisition boards.

11) Corrections to macro language compiler:

- string variable comparisons in "if-then" and "while"
- improved type-checking in the "a ? b" portions of the "if a ? b then", "while a ? b", and "for a ? b" constructs

12) Correction to device driver for Ariel DSP-32C boards, which was sometimes incorrectly sizing the total amount of memory on the board, producing a "data memory less than 1024K" message after sizing the DSP-32C board memory. The problem had to do with timing of the host machine and late-model revision (D, E, and F) DSP-32C boards. Also, automatic detection and utilization of the DSP-32C's 16-bit transfer mode has now been installed. Hypersignal-Macro will degrade to 8-bit mode if the DSP-32C is installed in an 8-bit slot or the 16-bit mode jumper on the DSP-32C is removed.

13) The constant values shown in section 7.2.4 (page 7-7 of revision 3.1) of the Users Manual are incorrect. The values should be as follows:

0

1

2

3

4

5

7

0

1

2

3.141529...

The constant FFTlen has been removed. If you need the FFT length of a waveform file, first access the FFT order, then calculate FFTlen as:

```
FFTlen = Ln(2)
FFTlen = FFT_order*FFTlen
```

Also, "FFTlen" as shown on page 7-35 of revision 3.1 should be ignored.

14) Writing time data traces from the Spectrum Analyzer display to (time domain) waveform files has been corrected.

15) Transfer function entry in the Spectrum Analyzer function has been changed so that display is ALWAYS the ratio of TRACE1/TRACE2. Thus, by entering 'Hnn+' in either the F1 or F2 fields, either the test-system output or input can be viewed simultaneously with the log magnitude of the system's frequency response. By reversing the physical channel entry in the TRACE1 or TRACE2 fields, the ratio can be reversed.

16) Support for revision B of the DSP Research "BB71" external analog I/O unit support added (to account for interface cable change made by DSP Research from revision A to revision B). Enter 'DRIC30-BB71' or 'DRIC30-BB71B' in System Configuration menu. Enter 'DRIC30-BB71A' for revision A systems.

17) System Configuration designators for the Ariel PC-56 and DSP-56 have been switched to be consistent with other Ariel PC- and DSP- board pairs. Use an 'ARI56' or 'ARI56-A' entry for the DSP-56, and an entry of 'ARI56-B' for the PC-56.

Release Notes, Hypersignal-Macro Series, FEB92-JUN92

Release notes for Hypersignal-Macro series software from February 1992 through June 1992 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.20a - v3.20e, Hypersignal-Acoustic v3.20a - v3.20e, and Hypersignal-Macro NRT v4.20a - v4.20e.

New Features, Hypersignal-Macro Series, FEB92-JUN92

1) New revision of documentation:

- rev. 3.1 of Users Manual
- rev. 3.0 of Hardware Reference Guide
- rev. 2.0 of Software Reference Guide
- rev. 2.0 of Tutorial

2) 'Shift-2' or 'Shift-1' can be used to place instrumentation displays in a mode where scaling and offset commands apply to both traces of a dual-trace display simultaneously, instead of only to the active trace.

3) 'O1' and 'O2' can be entered in the dual-trace control subfield of all applicable displays to display only one trace in overlaid-trace mode. Effectively, this allows dual-trace operation, but with only one trace displayed. For example, when viewing transfer function display in the Spectrum Analyzer, dual-trace operation is required, but it may be desirable to maximize the viewing area with only the transfer function

result and not any additional data format.

4) All DSP chip Source Code Interfaces updated to include both the register value needed for external clock and the actual sampling rate (in Hertz) as part of the shared variable list initially set by the host PC. This allows the high-level software to set a hardware-dependent value for external clock (if appropriate), and also set the desired sampling rate in Hz, allowing programmers access to this information in the DSP Source Code Interfaces. All interfaces now contain a value called "FS_MODE" for the hardware register value, and a value called "FS_VALUE" for actual sampling rate.

5) True "previous page" operation (using 'PgUp' key) added to Directory Display menu function.

6) Further improvements and enhancements to DSP96002 drivers, including consolidation of Ariel and Spectrum DSP code.

7) Addition of EXPONENT EXPAND field to the ASCII Export function, allowing the waveform data to be optionally multiplied by a scaling factor corresponding to the exponent stored in the waveform header. In this case, the exponent in the waveform header is then exported as unity (equal to 1). For floating-point export especially, this allows waveform data to be exported with no implicit overall exponent,

facilitating their use with other programs.

8) 32-bit integer "IOC register" values added to the configuration file for each DSP32C board supported, so that IOC register settings can be controlled at a high level, rather than changing DSP32C source code. See the "config.doc" file on Hypersignal-Macro Series software distribution diskette #2 for location of these values.

9) Addition of CAC AC5-series "DBD" dual 16-bit input and output daughterboard support.

10) New board driver finalized: Ariel DSP-96 board. Enter 'ARI96' in System Config menu.

11) New board driver added: "BB71" 4-channel analog I/O option added to DSP Research Tiger-30 board. Enter 'DRIC30-BB71' from System Config menu.

12) Block-floating point support installed in Difference Equations.

13) Spectrum (LSI) DSP96002 system board driver finalized. Enter 'LSI96' in System Config menu.

Bug-Fixes/Corrections, Hypersignal-Macro Series, FEB92-JUN92

1) Correction to Difference Equation stack problem when used in macro mode.

2) Correction to LSI DSP96002 FFT Generation hang-up problem on slower machines.

3) Correction to stereo convolution.

4) Change to high-level DSP96002 driver that eliminates need for PAL change when using later revision LSI DSP96002 boards. V3.20a through 3.20c worked with V3 board PALs and below, but not with V4 PALs. A software "work around" was possible that allowed the software to operate in both cases.

5) Correction to ASCII Export function when exporting and importing waveform data files with extended headers (proper number of header values were not being written out correctly).

6) Correction to FFT Generation operation in macro mode when using LSI ADSP-2101 board.

7) Correction to transfer function operation in the Spectrum Analyzer function with all TMS320C30-based boards.

8) Correction to use of Analog Conversion function from during macro record mode ("learn" mode). The menu command section was being written out before the "BEGIN ACQUISITION" prompt was answered, instead of afterwards, resulting in macro files that wouldn't "playback" correctly.

9) Correction to accelerated functions invoked from a macro file when using the LSI (Spectrum) TMS320C30 board. A reset timing

problem caused this board to produce invalid results in macro mode (accelerated functions only).

10) Correction to Ariel DSP-32C D/A output sampling rate problem with low sampling rates (less than 16 kHz).

11) Correction to transfer function display in Spectrum Analyzer for all boards.

12) Correction to LSI DSP96002 system board interrupt service routine context save problem (intermittent stack problem).

13) Major re-write of DSP96002 code generation for filters.

Release Notes, Hypersignal-Macro Series, DEC91-JAN92

Release notes for Hypersignal-Macro series software from December 1991 through January 1992 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.10 - v3.10d, Hypersignal-Acoustic v3.10 - v3.10d, and Hypersignal-Macro NRT v4.10 - v4.10d.

New Features, Hypersignal-Macro Series, DEC91-JAN92

1) Macro Manager menu now invoked with 'F3' key from main menu (and Directory Display function re-installed as a menu selection).

2) A record mode has been installed ('F4' key from main menu)

that allows the user to step through a sequence manually and have the program automatically generate the appropriate macro file for later use.

3) Block-floating-point exponent now used in both time domain waveforms as well as frequency domain waveforms. This allows Hypersignal to acquire and create waveforms that contain floating-point data ranging from to 3.90625e-03 to 4.194176e+06. Functions newly affected include File Acquisition, time domain functions, Inverse FFT, and Code Generation.

4) "Inverted filter output" function in the Spectrum Analyzer that writes out inverted magnitude traces in FIR filter specification file format. After designing the FIR filter to match the inverted trace, the filter can be applied in real-time to produce a "compensated" frequency response that is more or less flat. Hypersignal-Acoustic only.

5) Transfer function display in Spectrum Analyzer now shows 0 dB as center of trace, with +N and -N dB above and below, where N is controlled by the vertical scaling commands ('PgDn' and 'PgUp' keys)

6) Improvements to filter design, including:

- maximum length of Kaiser Window filters increased to 4096
- maximum length of Parks-McClellan filters increased to 769

- passband ripple value now must be entered for Steiglitz IIR filters (in dB, as with other design methods; a 0 value will cause existing designs to be unchanged)
- new, "two-level" error entry for Steiglitz IIR arbitrary method, which allows initial pole-zero pattern (initial convergence) to be established more easily
- impulse response waveform files created by the Filter Construction functions have been changed so that an FFT will produce a frequency response indicating the absolute position of 0 dB (unity gain), as opposed to a relative position. Thus, returning to the filter design display will show exact filter passband gain relative to 0 dB. Further implied by this change is the ability to create arbitrary filter designs with gain more or less than 0 dB, by creating the desired response on the new scale. This change has other ramifications for filter display throughout Hypersignal. A Magnitude Display of a filter response now shows 0 dB as being the same for autoscale or system calibration (default) modes of display. Also, the Spectrum Analyzer will show filter responses from file input correctly relative to 0 dB and to simultaneous analog input.
- a "tolerance region toggle" command has been installed in the graphical filter design functions ('T' key). Especially useful for arbitrary designs, this command can be used to show the original menu specifications for comparison with designs have been modified and/or their frequency responses.

7) "GetDateTime" function added to macro language.

8) New board driver added: CAC AC0 series, including dual ADC and codec daughterboards. Enter 'CAC32C-ADC' and 'CAC32C-CODEC' in System Config menu.

Bug-Fixes/Corrections, Hypersignal-Macro Series, DEC91-JAN92

1) Removal of 32767 FRAMESIZE limit in Difference Equations.

2) Correction to long-term problem in instrumentation displays where display traces using waveform files as input were not correctly written to disk by the WRITE function ('W' key).

3) Correction to long-term problem in accelerated convolution/correlation using TMS320C25-based boards where filters longer than 256 points were not applied correctly.

4) Correction to long-term problem in Spectrum Analyzer transfer function for DSP32C boards, which was not being calculated correctly.

5) The following macro language problems have been corrected:

- error message when a waveform file needed by a DSP or display function could not be found during macro execution was not correct
- menu functions missing an "endfunc" statement did not produce an error message that was always correct or indicated the correct line number
- assigning a Macro Manager field to a string variable (e.g. strvar = #1) did not function correctly
- some type conversion problems for real variables
- some "loopholes" in type checking, where some type mismatches were not detected, including a constant on the lefthand side of an "="
- parameters could not be passed to procedures by "value" (e.g. constants, Macro Manager menu references, @ function results)
- strings could not be concatenated (added together)
- assignment statement of form a = b + #N would compile, but a = #N + b would not

- strings returned as conversions from real numbers are now trimmed for leading spaces, trailing mantissa zeros, and leading exponent zeros, to make them suitable for use in Difference Equation menu fields
- automatic type conversion for output of the GetHeaderValue and GetDataValue functions was not being performed--the output of these functions can now be placed in either integer or real variables, but real variables are suggested for waveform values and for the "Max Amplitude" value from a GetHeaderValue(filename, Amp) call

6) A slight gain problem with IIR lowpass and highpass filters with even filter orders and IIR bandpass and bandstop filters with filter order / 2 equal to an even number has been corrected. The problem was ripple that could occur such that the minimum ripple was at 0 dB instead of the maximum ripple.

7) The tracking crosshair cursor in dual-trace waveform displays has been corrected in the following case: trace 1 shows a waveform whose total length is less than the display and also less than the trace 2 waveform, and the cursor would not read out trace 2 values.

8) Correction to 2-D Spectrograph print-out from monochromatic displays (such as Hercules board, or IBM CGA graphics adaptor).

9) Further correction to frequency domain displays resulting from complex time domain data, where every so many frames would display a zero frame, or uninitialized frame.

Release Notes, Hypersignal-Macro Series, JUL91-NOV91

Release notes for Hypersignal-Macro series software from July 1991 through November 1991 cover Hypersignal-Macro RT and Hypersignal-Macro EX v3.00 - v3.00g, Hypersignal-Acoustic v3.00 - v3.00g, and Hypersignal-Macro NRT v4.00 - v4.00g.

New Features, Hypersignal-Macro Series, JUL91-NOV91

1) Macro file language. Enter the name of a desired macro file to compile and execute in the "Macro Manager" menu selection. Entering the last field or pressing 'Ctrl-Enter' in this menu will cause the macro to compile if necessary, and execute. Also, macro files can be executed remotely by another program, by passing the macro name in the Hypersignal command line. Macro language files allow all Hypersignal functions to be invoked (and controlled by "passing" menu parameters), "if-then-else" constructs, loops, procedures, variables (integer, real, string) that can be manipulated and then used as input to menu fields in order to control Hypersignal operation dynamically, scheduling (using system clock), math functions, and many other built-in commands and functions. By passing "parameters" to a macro via the Macro Manager menu, macros

become re-usable, functioning as "Hypersignal procedures". An excellent example of this is illustrated in the macro file "SAMCONV.MAC", included on distribution diskette #2. SAMCONV.MAC converts a waveform sampling frequency either up or down, by invoking Difference Equations, FIR Filter Construction, and Convolution in the correct sequence required to "interpolate" or "decimate". Simply by changing the input and output waveform names and "stretch" and "contract" factors that make up the input parameters, this macro can be used over and over without and further editing. Essentially, the Macro Manager menu becomes a "virtual menu" for many different functions. To further enhance the virtual menu concept, the menu labels can also be customized for each macro.

2) New Users Manual. Completely reorganized, revised and updated.

3) Completely revised on-line Help. Reduced in size by 1/3, as much of detailed, non-essential information has been placed in Users Manual. Overall, the on-line Help has been made much easier to use, more concise, and more focused on information needed "at the moment" when using the software.

4) Updated Hardware Reference Guide. Updated for some existing boards and for all new boards.

5) Vertical limit (top and bottom display values) control in frequency domain display functions; either autoscale (zero entry) or explicit entry possible.

6) Explicit entry of vertical limits in frequency domain displays (see 5, above) causes single/dual waveform displays to be displayed relative to a "reference dB value", which can be set via the Spectrum Analyzer analog input calibration function or in the configuration file. The default value assigns 32767 (full-scale input from a 16-bit A/D converter) to 0 dB.

7) Dual-trace crosshair cursor with frequency and linear/log magnitude read-out installed in Spectrum Analyzer function.

8) Calibration function has been installed in real-time Spectrum Analyzer that sets the system "reference dB value" and the engineering units to be assigned to this value (in dB). Use the up and down arrow keys or mouse to position the crosshair cursor on the peak of the reference tone and press the 'E' key ("engineering units"). A prompt will appear asking you for the value to use in dB.

9) Overlaid waveform display in Waveform Display/Edit and Digital Oscilloscope (time domain), and Magnitude Display, Phase Display and Spectrum Analyzer (frequency domain) functions. Enter 'O' in the FORMAT,PLOT field for overlaid dual-waveform display and 'D' for dual waveform display (the

default). For Digital Oscilloscope, enter 'O' or 'D' in the first part of the FM,PLT,GRD field. For Spectrum Analyzer, enter 'O' or 'D' in the second part of the TPL field. Overlaid display is not active with map mode.

10) Addition of "limited imaginary" operator 'j' to Difference Equations. Equations can be entered in form a+jb or a+j*b, causing an imaginary component to be generated and the resulting output waveform to be complex. Currently, this is the only form in which 'j' is allowed; in other words, the program does not treat it as a true mathematical constant. For example, SQR(j) will not produce -1.

11) File Acquisition function modifications:

- complex waveform input is now allowed
- u-Law or A-Law expansion of incoming data can be achieved via an 'L' or 'A' attribute entry after the FORMAT field entry

12) Crosshair-cursor with frequency read-out has been added to 2-D Spectrograph display function. Linear or log frequency axis read-out.

13) A "mark section" command has been added to Waveform Display/Edit function. Use of the 'M' key writes out target window edges (or display edges if window not shown) to ASCII "mark" file. Mark files can contain sequence of marked waveform sections for macro or external processing.

14) Several acoustic-audio enhancements, including:

- minimum log frequency display value (low-end cutoff) control in all frequency domain display functions, including 2-D Spectrograph
- graphical display and functional improvements to 1/N octave band display
- Spectrum Analyzer calibration and engineering units (see 8, above)
- Log amplitude display option in Waveform Display/Edit function, including both data-entry menu field and log/linear toggle command inside display ('S' key)
- Schroeder integration macro that produces time domain output (Hypersignal-Acoustic only)
- stand-alone utility that reads SYSid complex output files (Hypersignal-Acoustic only)
- snap-in real-time filtering can apply arbitrary time domain files as filters--intended for "room auralization" convolution in real-time (Hypersignal-Acoustic only)

15) Snap-in filtering in Digital Oscilloscope, Spectrum Analyzer, and Analog Conversion now look for a time domain waveform (.TIM extension) if no standard .IIR or .FIR filter coefficient file is found. Thus, these functions can now be used also to convolve arbitrary impulse responses with incoming analog data in real-time on the DSP board.

16) Analog Conversion continuous D/A output from disk "infinite loop" option. Enter 'GL' to cause output waveform file to repeat until until you press 'ESC' key.

17) Six times faster downloading for DSP boards using COFF file

format (DSP32C, TMS320C30, TMS320C25).

18) Relative waveform file paths can now be used (DRIVE entry blank, and SUBDIRECTORY entry = ".").

19) New board drivers added:

- DSP Research Tiger-30. Enter 'DRIC30' in System Config menu.
- Techron TEF-20. Enter 'TEF20' in System Config menu (Hypersignal-Acoustic only).
- Ariel PC-32C with ProPort. Enter 'ARI32C-b' in System Config enu (Hypersignal-Acoustic only).
- Burr-Brown ZPB34 and ZPB100 speech board combination. Enter 'ZPB34-ZPB100' in System Config menu.

Bug-Fixes/Corrections, Hypersignal-Macro Series, JUL91-NOV91

1) Further modification to LSI TMS320C30 device driver that corrects unreliable operation of the LSI TMS320C30 system board when used with the LSI memory expansion board.

2) Further modification to frequency domain displays when displaying FFT output data that resulted from complex time domain input data.

3) Correction to LSI ADSP-2101 driver that caused hangup when R/I data is requested in the FFT Generation function.

4) Correction to LSI ADSP-2101 driver problem where overflow was sometimes not detected in FFT

Generation function, and resulting frequency domain files were garbage.

5) Correction to NEC 77220 driver that caused problems with iFFT results. Also, correction to NEC 77220 source code for forward FFT processing, and overflow detection.

6) Full activation of trace 1/2 selection in Magnitude Display and Phase Display functions. Continued improvement to trace 1/2 options in Waveform Display/Edit function.

7) Correction to some dual-trace waveform editing problems with stereo and complex waveforms in the Waveform Display/Edit function.

8) Change to Digital Oscilloscope offset function (up and down arrow keys) so that y-axis amplitude units track correctly with vertical waveform movement. The offset function has also been activated for time-domain traces in the Spectrum Analyzer function. All other data formats in Spectrum Analyzer (non-time formats) show crosshair cursor in response to up/down arrow keys or mouse.

9) Minor corrections to DSP96002 code generation.

10) Fix to long-time bug in Spectrum Analyzer where EMS print-outs would cause run-time error 207.

11) Fix to long-time bug in Difference Equations caused by uninitialzed variable. Symptom

was rare, random run-time error 207 for no apparent reason.

12) Fix to long-time bug in FIR Filter Construction where software, on rare occasion, would freeze up after displaying "Generating Parks-McClellan Coefficients" message.

3.3.2 Revision History for DSPower-HWLib Software

NOTE: Use the Browse (">>" and "<<") buttons to scroll forward and backwards in the release notes.

Most Recent Release Notes

New Features, Hypersignal-Macro Series, MAR95

Bug-Fixes/Corrections, Hypersignal-Macro Series, MAR95

New Features, DSPower-HWLib, MAR95

Bug-Fixes/Corrections, DSPower-HWLib, MAR95

New Features, DSP Source Code Interface, MAR95

Bug-Fixes/Corrections, DSP Source Code Interface, MAR95

New Features, DSPower-Block Diagram, APR95

Bug-Fixes/Corrections, DSPower-Block Diagram, APR95

Release Notes, DSPower-HWLib, MAR95

Release notes for DSPower-HWLib software in February 1995 through present cover DSPower-HWLib v1.1

New Features, DSPower-HWLib, MAR95

1) Added Microsoft Visual C/C++ 1.5 project files (.mak) and WATCOM C/C++ v10.0 project files (.wpj).

2) Subdivision of engine functions into enmgr.dll and renaming of hwsel.dll to hwmgr.dll. This organization is in preparation for first release of splib.dll, which is DSPower-SPLib (signal processing library) software. Error message also reorganized between enmgr.h, hwmgr.h, and hwlib.h include files.

3) Added DSExecEngineFunction call to invoke arbitrary engine functions. Included execfunc.cpp source code example to illustrate use of this call.

4) Changed format of DSCalcSampFreq call to account for differences in Borland C/C++ v4.5 and Microsoft Visual C/C++ v1.5 compiler in returning a double from a function.

5) Renaming of DSEnableStatusWindow and DSDisableStatusWindow calls to DSShowEngMgrStatusWindow and DSHideEngMgrStatusWindow. DSAddEngMgrStatusLine now returns the window handle to the status window.

Bug-Fixes/Corrections, DSPower-HWLib, MAR95

1) Corrected internal hwmgr.dll problem concerning location of

hwsetup.lst path. This problem was preventing user programs from being run from arbitrary locations other than \dspower\hwlib.

2) Made DSP Engine timeouts variable depending on nature of DSP Engine calls. Increased DSP Engine timeout value for DSEngineOpen call, to account for reports of timeout on 25 MHz 386 machines.

3.3.3 Revision History for DSPower-SPLib Software

This section not available yet.

NOTE: Use the Browse (">>" and "<<") buttons to scroll forward and backwards in the release notes.

Most Recent Release Notes

New Features, Hypersignal-Macro Series, MAR95

Bug-Fixes/Corrections, Hypersignal-Macro Series, MAR95

New Features, DSPower-HWLib, MAR95

Bug-Fixes/Corrections, DSPower-HWLib, MAR95

New Features, DSP Source Code Interface, MAR95

Bug-Fixes/Corrections, DSP Source Code Interface, MAR95

New Features, DSPower-Block Diagram, APR95

Bug-Fixes/Corrections, DSPower-Block Diagram, APR95

3.3.4 Revision History for DSP Source Code Interface Software

There are two versions of DSP Source Code Interface software: source code math and DSP functions, ISRs, and operating modules necessary to provide Hypersignal-Macro series functionality, and source code that provides additional functionality for other Signalogic DSP Software components. For a revision history list of DSP Source Code Interface software through March 1995, see section 3.3.1 above, Revision History for Hypersignal-Macro Series Software. For revisions after this point, see this section, in which each release note section indicates which Signalogic DSP Software components are affected.

Notes

The version number of DSP Source Code Interface software starts with 5.0 below. This is an arbitrary decision made necessary because changes no longer necessarily track exactly with releases of Hypersignal-Macro series software.

Revisions affecting Hypersignal functionality also apply if Hypersignal

is being used as a DSP Engine, for example by DSPower-HWLib or DSPower-Block Diagram software.

Changes are not necessarily made to all supported DSP devices at the same time. Revisions should indicate which devices are affected.

An NDA is usually required for use of the DSP Source Code Interfaces. See section 1.2.3, DSP Source Code Interface Software Summary, and chapter 4, Using DSP Source Code Interface Software, for more information.

NOTE: Use the Browse (">>" and "<<") buttons to scroll forward and backwards in the release notes.

Most Recent Release Notes

New Features, Hypersignal-Macro Series, MAR95

Bug-Fixes/Corrections, Hypersignal-Macro Series, MAR95

New Features, DSPower-HWLib, MAR95

Bug-Fixes/Corrections, DSPower-HWLib, MAR95

New Features, DSP Source Code Interface, MAR95

Bug-Fixes/Corrections, DSP Source Code Interface, MAR95

New Features, DSPower-Block Diagram, APR95

Bug-Fixes/Corrections, DSPower-Block Diagram, APR95

Release Notes, DSP Source Code Interface, MAR95

Release notes for DSP Source Code Interface software in March 1995 through present cover v5.0.

New Features, DSP Source Code Interface, MAR95

1) Added dynamic MLS sequence generation to DSP5600x source code. Affects Hypersignal version.

2) Added programmable output attenuation capability to Sig32C-8 board DSP32C analog I/O code. Affects Hypersignal version.

3) Added first version of TMS320C5x source code; initial support is for Innovative Integration PC50 board. Affects Hypersignal version.

Bug-Fixes/Corrections, DSP Source Code Interface, MAR95

1) Corrected gain problem in Sig32C-8 board DSP32C analog setup code. Affects Hypersignal version.

2) Added dead-start calibration routines to Tahiti board DSP5600x analog setup code. Affects Hypersignal version.

3.3.5 Revision History for DSPower-Block Diagram Software

NOTE: Use the Browse (">>" and "<<") buttons to scroll forward and backwards in the release notes.

Most Recent Release Notes

New Features, Hypersignal-Macro Series, MAR95

Bug-Fixes/Corrections, Hypersignal-Macro Series, MAR95

New Features, DSPower-HWLib, MAR95

Bug-Fixes/Corrections, DSPower-HWLib, MAR95

New Features, DSP Source Code Interface, MAR95

Bug-Fixes/Corrections, DSP Source Code Interface, MAR95

New Features, DSPower-Block Diagram, APR95

Bug-Fixes/Corrections, DSPower-Block Diagram, APR95

Release Notes, DSPower-Block Diagram, APR95

Release notes for DSPower-Block Diagram software in March 1995 through present cover DSPower-Block Diagram v1.1

New Features, DSPower-Block Diagram, APR95

1) Added improved toolbar interfaces to instrument and display blocks.

3) Added more fly-over Help features, including context-sensitive Help button.

4) Added worksheet toolbar preference options (standard MDI, vertical, floating).

5) Added option for up to eight (8) system library and user library locations to be in use.

Bug-Fixes/Corrections, DSPower-Block Diagram, APR95

1) Corrected numerous diagram saving and retrieving problems.

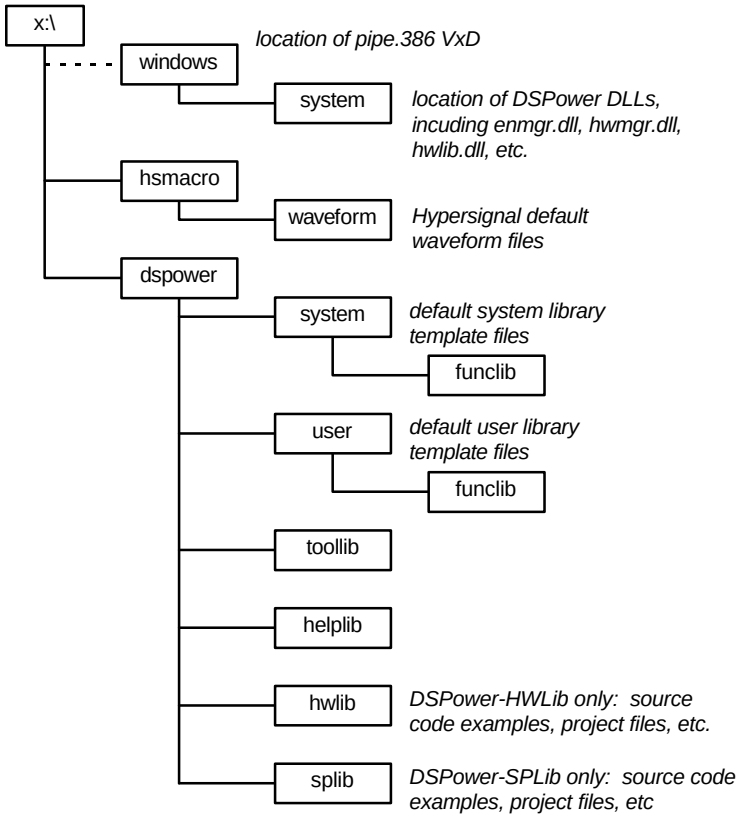
3.4 Directory Structure Chart

In the directory structure chart below, Signalogic DSP Software components are shown as they are stored in general relationship to one another. There can be many exceptions to this; for example, the name "dspower" is only a default suggestion. Also, "hsmacro" should be replaced by "hsmacnrt", "hsacous", etc. depending on which Hypersignal-Macro series package was installed, if any (see section 3.2.5, Installing and Configuring Hypersignal for User Mode Operation).

The root drive is shown as "x:\", which would be "c:\", "d:\", a network drive, etc. after actual installation. Also, the "windows"

subdirectory may not be located on the same drive as the Signalogic software.

Note that DSPower-HWLib and DSPower-SPLib components are installed separately, a "dspower" subdirectory is still created, although other segments of this subdirectory having to do with DSPower-Block Diagram may not be created.



General Topics

Media and Disk Space Requirements

Documentation and Media You Should Have Received
Before You Install
Installing Hypersignal-Macro Series Software
Installing DSPower Series Software

3.5 Uninstalling

This section not available yet.

4 Troubleshooting

Overview

This section is a loosely organized collection of fixes and workarounds to common problems that can be encountered when using Signalogic DSP Software and supported DSP/analog hardware. Each entry is presented as a problem description, followed by one or more suggestions for relief.

If you have read what there is to read, and then consulted this

section, and your problem still doesn't go away, please try the following:

1. send e-mail to:

   DSPhelp@signalogic.com

   or

   72113.1412@compuserve.com

2. send fax to 214-343-0163

3. if absolute urgency is required, contact Signalogic technical support at 214-343-0069

NOTE: Technical support calls to Signalogic 1-800 numbers will not be accepted.

Main Topics

Troubleshooting Installation Problems

Troubleshooting Video and Display Problems

Troubleshooting DSP/Analog Hardware Problems

I/O-Mapped Hardware Conflicts

Memory-Mapped Transfer Gotchas

Host PC Internal Bus Speed Conflicts

4.1 Troubleshooting Installation Problems

This section is a loosely organized collection of fixes and workarounds to common problems that can be encountered when installing Signalogic DSP Software. Each entry is presented as a problem

description, followed by one or more suggestions to either correct the problem or workaround it.

Problem: during installation, I see a box saying "file to be installed is older than existing file; keep existing file?"

A dialog box with this message, and showing statistical data from two different files with the same filename is displayed during installation of a DSPower series software package.

Suggestion(s)

Keep the existing file. What has probably happened is a common library or DLL used by many software packages and needed by DSPower software, such as a Borland or Microsoft control library, already exists on your computer system. If such a case, this type of dialog box is displayed if the existing version is actually newer than the version the Signalogic Software Setup Utility is trying to install. This can happen if you have obtained a more recent copy of the common library by upgrading, buying a new system, etc.

Problem: when starting Windows on my system, I see a message "PIPE Virtual Device Driver, v3.07"

After installing DSPower-Block Diagram, or DSPower-HWLib, this message appears on your screen when first starting Windows.

Suggestion(s)

Ignore this message. Version 3.07 of the Pipe.386 virtual device

driver (VxD) displays a sign-on message which has no effect. This VxD is used to allow DSPower-Block Diagram to communicate with Hypersignal packages running transparently in the background, and is installed during the normal course of DSPower-Block Diagram or DSPower-HWLib installation. For more information about Pipe.386 VxD, see Signalogic DSP Software System Guide section 1.1, Relationship of Signalogic DSP Software Components.

Problem: when running a DSPower-Block Diagram function, I get a message "DSP Engine not found", or similar

After clicking on the Run button inside a function or display block, or attempting to run a block diagram, a "DSP Engine not found" or "DSP Engine not invoked" message box appears, and the individual block or overall block diagram does not execute.

Suggestion(s)

There a number of ways this can happen:

1. The DSP Engine software was not installed correctly.

This is most likely to happen if Hypersignal was installed separately, not using the Signalogic Software Setup Utility. If Hypersignal has been installed separately (a process consisting mainly of copying *.* from distribution diskettes 1 and 2 to an "hsmacro" or "hsacous" subdirectory, and *.* from diskette 3 to an "hsacous\waveform" or "hsmacro\waveform" subdirectory), Hypersignal must be run at *least once* in stand-alone mode. This is necessary to expand several files distributed in compressed form on the Hypersignal distribution diskettes. This expansion occurs automatically if Hypersignal is installed using the Signalogic Software Setup Utility.

2. The DSP Engine path is not configured properly inside DSPower-Block Diagram. Ensure that under the DSPower-Block Diagram main menu selection "Configuration", and submenu selection "DSP Engines", that you have entered both a valid command line and run-time source code path for any DSP Engine packages currently in use. The run-time source code path entry should contain a leading drive specifier, and no trailing backslash; for example:

..Engine Command Line **c:\hsmacro\hsmacro.exe**
Engine Run-Time Source Path **c:\hsmacro**

3. The version of DSP Engine software is not current. For example, you should have at least version 4.00xf or higher of Hypersignal-Macro, Hypersignal-Macro EX, or Hypersignal-Acoustic, version 5.00xf or higher of Hypersignal-Macro NRT, or version 4.00 or higher of MATLAB combined with version 3.0b or higher of the Signal Processing Toolbox. See section 1.3, System Requirements.

Problem: the DSPower-HWLib demo programs give a message "hardware not responding" and then fail to run

After clicking on the Digital Oscilloscope, Digital Tape Recorder, or other DSPower-HWLib demo icons from the Signalogic DSP Software group in Program Manager, a message box is displayed indicating "DSP hardware not responding" or "DSP hardware failed to initialize" or similar.

Suggestion(s)

Such a message probably means the DSP/analog hardware is not properly installed, or does not have a correct I/O base address or memory base address set in the Hardware Selector. Re-run the demo and double-check the address values in the Hardware Selector dialog box (lower right) vs. the jumper or switch settings on the DSP/analog hardware. If the demo

program still does not run, there may be a more serious problem. See section 4.3, Troubleshooting DSP/Analog Hardware Problems, for more detailed information.

Problem: in my Hypersignal update (or upgrade), I only received diskettes 1/3 and 2/3. Why not diskette 3/3?

Assuming that you already had a copy of a Hypersignal-series software package, you may have received an update of a new version or upgrade to a more powerful package that did not include diskette #3 out of 3.

Suggestion(s)

This can occur because diskette #3 contains only waveform files, and typically does not change between version updates. It is also the same for all levels of Hypersignal-series packages. If you have received a Hypersignal package for the first time, you should definitely have received diskette #3.

Problem: after installing a Hypersignal-Macro series package, running the hs.bat, hsm.bat, or hsa.bat files doesn't work

This is most likely to occur when installing the Hypersignal software without using the Signalogic Software Setup utility. It is possible that the hsx.bat file failed to create a HSxxxxx.OVR file, when being run for the first time after installations, because the hsx.bat file failed to locate a copy of "EXPAND.EXE" on your system..

Suggestion(s)

After following the normal Hypersignal installation procedure outlined in section 3.2.5, Installing and Configuring Hypersignal Software For User Mode Operation, you should locate a copy of EXPAND.EXE on your system (Microsoft, version 2.0 or higher), select your subdirectory to wherever Hypersignal has been installed, and type:

```
X:\yyy\expand
HSxxxxx.ov_ Hsxxxxx.ovr
<Enter>
```

at the DOS command line to expand the file, where X:\yyy is the location of EXPAND.EXE and xxxxx is macro, acous, macnrt, etc. depending on the type of Hypersignal-Macro series package being installed. EXPAND.EXE is shipped standard by Microsoft with MS-DOS versions 5.0 and higher.

The above procedure should also be followed on any files with .ex_ extension, that did not get expanded to their .exe equivalents; use a

```
dir *.ex_
```

command to check for any other unexpanded files.

NOTE: When installing a Hypersignal-Macro series package using the Signalogic software Setup Utility, compressed files are automatically expanded as they are copied from the distribution floppy diskettes to your hard disk.

4.2 Troubleshooting Video and Display Problems

This section is a loosely organized collection of fixes and workarounds to common problems associated with video drivers and display in both DOS and Windows 3.x. Although presented in the context of using Hypersignal-Macro series or DSPower series software, most of these problems are general to operation of any software in a DOS or Windows 3.x based system. Each entry is presented as a problem description, followed by one or more suggestions to either correct the problem or workaround it.

Problem: video display goes fuzzy

Video display goes "fuzzy" or loses sync when I run a Hypersignal package from Windows, and then return to Windows.

Suggestion(s)

1) Try installing the video driver supplied by the manufacturer of the your system's video hardware. The standard "SuperVGA" driver supplied with Windows is known to have this problem with some types of video hardware.

2) Try altering the settings in the hsa.pif or hsm.pif files, using the "PIF Editor" found in the Main group in Program Manager. Try adding the following settings, one by one:

- Video Memory: High Graphics
- Monitor Ports: High Graphics
- Retain Video Memory

Problem: Hypersignal display / instrument background not erased

When running Hypersignal from Windows icons or from a DOS prompt started from Windows, sometimes the screen background will not be fully erased in a display or instrument.

Suggestion(s)

Inside the Hypersignal-Macro series software display or instrument, try the 'Refresh' command (ctrl-L) from the keyboard. Versions 4.00bx and later of Hypersignal contain special "force clear" code for some graphics boards that have demonstrated this problem.

Problem: DSPower-Block Diagram objects too big

Your display is probably in 640x480 mode, and not enough screen resolution is available for default DSPower displays.

Suggestion(s)

1) Try setting the "Object Size Independent of Display Resolution" checkbox in the Worksheet Options menu. Note, however, that this option can make bitmaps appear "sketchy", or fuzzy, because the bitmap must be compressed to fit in a smaller area.

2) Try configuring your video board for 800x600 display by selecting the Windows Setup icon from the Main group in Program. Choose "Change System Settings" and look for a video driver option exactly the same as the one you are using, except with 800x600 resolution (i.e. same hardware name, same number of colors, same size font).

4.3 Troubleshooting DSP/Analog Hardware Problems

When using plug-in DSP/analog or acquisition hardware in a PC environment, there are some general areas which typically cause trouble:

The hardware uses only an I/O-mapped address, but seems to conflict with other hardware.

The hardware uses a memory-mapped addressing (data transfer) scheme, and seems to conflict with other hardware, already-installed software utilities, or the system itself.

PC bus speed (or internal system bus speed).

Problems during data acquisition or signal output.

4.3.1 I/O-Mapped Hardware Conflicts

The large majority of plug-in PC boards use an I/O-mapped address to enable software drivers to talk to them. The obvious problem is what happens when you get two or more boards, from different manufacturers, plugged into the same system--and they conflict, or occupy part of the same I/O-mapped address space.

After installing your DSP/analog hardware, here are some symptoms to look for if you suspect an I/O-mapped address conflict with other hardware:

- when being used in a software function, the DSP/analog hardware seems to run for a while and then freezes up, or gets through part of a function and then locks up
- data spikes or glitches are evident in digital oscilloscope or spectrum analyzer instrument displays
- another, unrelated board already in your system suddenly develops a problem
- otherwise strange or intermittent system behavior that was not present before installation of the DSP/analog hardware

If one of these symptoms occurs during check-out of your DSP/analog hardware, our best suggestion is to do as follows:

exit the software package, and exit Windows (if applicable)

"park" your hard disk (if applicable)

wait a few seconds to make sure all disk activity has ceased

turn the computer off

remove the DSP/analog hardware

change the DSP/analog hardware base I/O address by incrementing upwards at least 20H. This is typically accomplished using a bank of switches or set of jumpers on the board.

re-install the hardware, and try running software checks again

Why change the DSP/analog hardware base address, and not that of already installed board(s)? Generally, since DSP/analog hardware is intended for technical users in the first place, its documentation explains clearly and forthrightly how to change base addresses. Whereas with consumer-oriented boards, even though changing the base address is (must be) possible, it's often hard to find in the documentation, or is not well-explained.

A smaller percentage of DSP/analog hardware also use memory-mapped addressing. For a thorough discussion of possible problems that can occur with such

hardware, see section 4.3.2, Memory-Mapped Transfer Gotchas.

Related Troubleshooting Topics

Troubleshooting Installation Problems

Troubleshooting Video and Display Problems

Troubleshooting DSP/Analog Hardware Problems

I/O-Mapped Hardware Conflicts

Memory-Mapped Transfer Gotchas

Host PC Internal Bus Speed Conflicts

4.3.2 Memory-Mapped Transfer Gotchas

This section is only applicable if your DSP/analog hardware uses a memory-mapped transfer scheme (see individual hardware sections in Hypersignal-Macro Series Hardware Reference Guide, chapter 3, under "Digital Specifications".

Two 64 kbyte segments of the PC's "high memory", between 640k and 1024k, have traditionally been reserved for application-specific plug-in boards and/or memory-resident software utilities. The starting address for these segments is usually D000:0000H and E000:0000H. However, over time, such a plethora of boards and utilities have been created which require significant amounts of this limited resource, that conflicts can easily arise on any given PC. Most DSP board manufacturers try to

avoid using memory-mapped schemes, to save themselves the technical support headaches, but the temptation of somewhat better performance (and sometimes easier design) is strong, and there are still boards out there which use this technique. If you have DSP/analog hardware which uses the memory-mapped transfer mechanism, here are some symptoms to watch out for after installing the hardware:

- hardware seems to run for a while and then freezes up; hardware gets through part of a function and then locks up
- data spikes or glitches are displayed in functions like the digital oscilloscope or spectrum analyzer
- another, unrelated board or previously working software utility suddenly develops a problem
- otherwise strange or intermittent system behavior that was not present before installation of the DSP/analog hardware

If one of the above symptoms seems to be occurring, you may have encountered a memory-mapped gotcha. If so, here are some things you can check:

EMM386.EXE Considerations (Reserving Memory Areas)

The first thing you should be aware of is that if a line similar to

```
device =
C:\DOS\HIMEM.SYS
```

or

```
device =
C:\WINDOWS\HIMEM.SYS
```

appears in your CONFIG.SYS file, then any utility or program invoked afterwards could be loaded so as to conflict with memory needed by the DSP/analog hardware. This is because HIMEM.SYS makes it possible to load other programs in the high memory area mentioned above. If the HIMEM.SYS line is not present, you should add it. This line should appear before any other program or utility is invoked.

Look in your CONFIG.SYS file for an "EMM386" line. You may have something like

```
device =
C:\WINDOWS\EMM386.EXE
```

or

```
device =
C:\DOS\EMM386.EXE
```

followed by various memory usage specifications. The emm386 program controls PC access to high memory. It also controls access and usage of "expanded memory", which is an antiquated technique used for peripheral devices or software utilities which need a substantial amount of extra memory (i.e. more than .25 Mbyte). If this line is not in your CONFIG.SYS file, you should add it. Then, to reserve access to your memory-mapped DSP/analog hardware, try adding the following to this line:

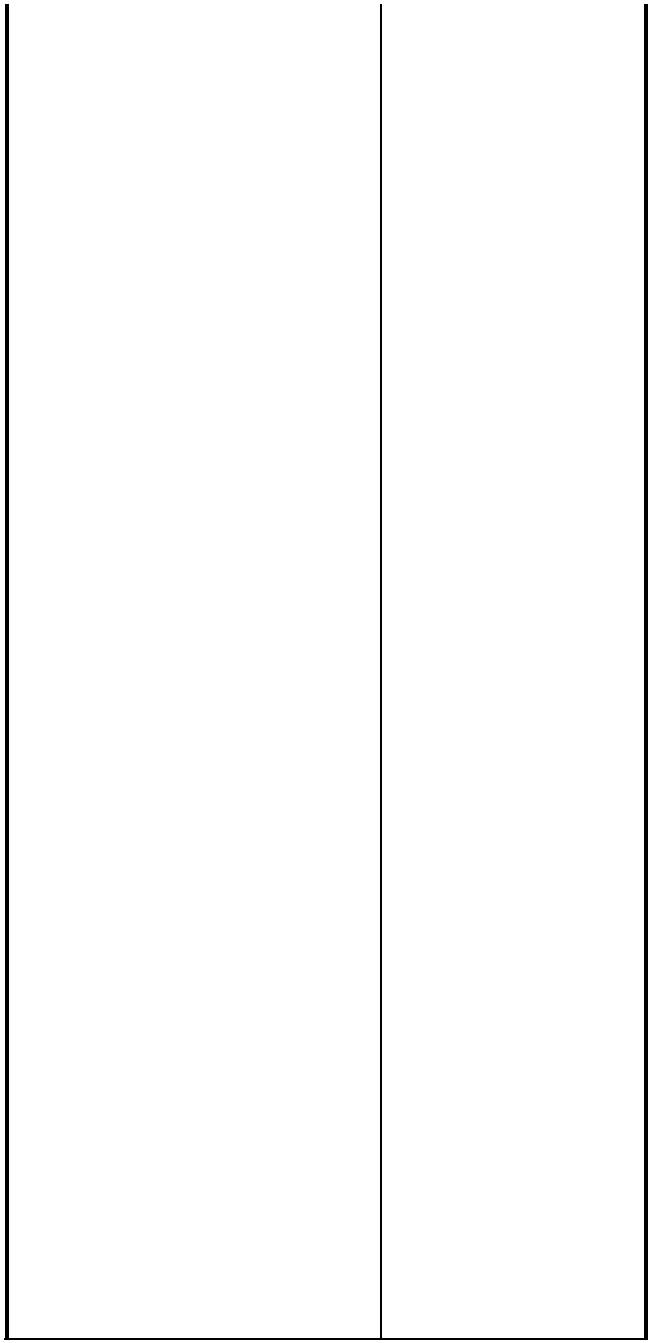
```
X=AAAA-BBBB NOEMS
```

Only add "NOEMS" if not already there; it must be *after* the "X=" command. If you are adding the

EMM386 line for the first time, put a space after EMM386.EXE. The "X=" command specifies a certain section of high memory be reserved. AAAA should be the starting address of the memory needed by your DSP/analog hardware, and BBBB the last address needed. Some examples (partial list only):

Table 13 Example CONFIG.SYS File EMM386 Line Entries

Hardware	EMM386 line addition



See the individual hardware sections in the Hypersignal-Macro Series Hardware Reference Guide, chapter 3, for memory-mapped base addresses of DSP/analog hardware that use this technique. Note that the addressing modes required by individual hardware are listed in the "Digital Specifications" table in the section for that hardware, under "Host Data Transfer Method".

NOTES: It is important the "EMM386" line appear immediately after the HIMEM.SYS line. This will ensure that not only does EMM386.EXE prevent other programs from being loaded into the reserved area, but EMM386.EXE itself does not load into the reserved area.

"NOEMS" specifies no expanded memory; if you think your system is using expanded memory, then the situation is more complicated. This would be the case if the EMM386.EXE line is already in your CONFIG.SYS file and it does not contain the NOEMS or X= commands. In this case, you should not add NOEMS to the EMM386.EXE line. You will need to figure out which segment of high memory your expanded memory is using, and set up the DSP/analog hardware to avoid this segment. For example, if your expanded memory is using segment

D000:0000, then you should change the base address on your DSP/analog hardware from Dxxx to Exxx (leave the lower digits the same), and then add an X=Eaaa-Ebbb command to your EMM386.EXE line, following the guidelines explained above.

You can also add a "V" to the end of the EMM386.EXE line to cause the status and results of the emm386 program to be printed onscreen when your computer is booting up.

Related Memory-Mapped Transfer Topics

EMM386.EXE Considerations (Reserving Memory Areas)

EMMExclude Lines in Windows SYSTEM.INI File

Memory-Mapped Hardware Conflicts

Software Conflicts (Utilities, Memory-Resident Programs)

EMM386.EXE: Microsoft Knowledge Base Excerpts

EMMExclude Lines in Windows SYSTEM.INI File

Look in your SYSTEM.INI file (found on \WINDOWS\SYSTEM subdirectory) for "EMMExclude=" lines, located in the "[386Enh]" section of this file. If this section exists, you should add an EMMExclude lines which matches up with the EMM386 command you have entered in your CONFIG.SYS file. If, for example, your CONFIG.SYS file contains an "X=D000-DFFF" command, then

you would add an "EMMExclude=D000-DFFF" line in the EMM section. (Actually, Microsoft documentation indicates that "X=" commands in the "EMM386" line take priority over EMMExclude lines in the SYSTEM.INI file, but we think it always a good idea to be extra safe with DOS-Windows interaction).

Related Memory-Mapped Transfer Topics

EMM386.EXE Considerations (Reserving Memory Areas)

EMMExclude Lines in Windows SYSTEM.INI File

Memory-Mapped Hardware Conflicts

Software Conflicts (Utilities, Memory-Resident Programs)

EMM386.EXE: Microsoft Knowledge Base Excerpts

Memory-Mapped Hardware Conflicts

Look for other boards in your system which may be using the same high memory segment. Typical culprits include network cards (which almost always require a small section of either the D000:0000 or E000:0000 segments), some high-performance video cards, other DSP/analog boards or sound boards (of course!), and non-standard peripheral cards (anything off the beaten track). If you find one, first simply try removing it before doing anything else. If the DSP/analog hardware problems clear up, then one of the contenders will need to have its base memory address changed. If you change the

DSP/analog hardware base address, be sure to inform the Hypersignal System Config menu, or the DSPower-Block Diagram DSP/Analog Hardware Configuration dialog box (see individual hardware sections in Hypersignal-Macro Series Hardware Reference Guide, chapter 3).

Related Memory-Mapped Transfer Topics

EMM386.EXE Considerations (Reserving Memory Areas)

EMMExclude Lines in Windows SYSTEM.INI File

Memory-Mapped Hardware Conflicts

Software Conflicts (Utilities, Memory-Resident Programs)

EMM386.EXE: Microsoft Knowledge Base Excerpts

Software Conflicts (Utilities, Memory-Resident Programs)

Look for software utilities or memory-resident programs which might be using high memory. You can do this by using the "mem" or "mem /s" commands from the DOS command line. Occasionally, such programs can be so ill-behaved they even use areas reserved by the EMM386.EXE program (see above). In such cases, you have no choice but to temporarily "REM-out" these utilities one-by-one from your CONFIG.SYS or AUTOEXEC.BAT files until you find out what is causing the problem. Sometimes, you can simply change the order in which these utilities are invoked to make a

problem go away--loading ill-behaved programs later increases the chance hey will use areas reserved by EMM386.EXE, since other (well-behaved) programs have loaded themselves with due respect to areas reserved by EMM386.EXE.

NOTE: Changing the order of utilities and programs in your CONFIG.SYS and AUTOEXEC.BAT files can be a risky business. Change the order only of those suspected of causing problems. These should not be well known utilities such as SMARTDRV.EXE, SHARE.EXE, DOSKEY.EXE, DOSSHELL.EXE, HIMEM.SYS, ANSI.SYS, etc.; they are much more likely to be programs supplied by third-party manufacturers, or having to do in some way with peripheral cards you have added to your system.

Related Memory-Mapped Transfer Topics

EMM386.EXE Considerations (Reserving Memory Areas)

EMMExclude Lines in Windows SYSTEM.INI File

Memory-Mapped Hardware Conflicts

Software Conflicts (Utilities, Memory-Resident Programs)

EMM386.EXE: Microsoft Knowledge Base Excerpts

EMM386.EXE: Microsoft Knowledge Base Excerpts

Below are some excerpts from files gathered from the Microsoft

Knowledge Base on the CompuServe network (all excerpts COPYRIGHT MICROSOFT 1994). These files should indicate the same information as presented above, but we've included them because they offer some extra detail which could prove useful in certain situations.

EMM386 Troubleshooting and Technical Information

Copyright Microsoft, 1994
Title: EMM386 Troubleshooting and Technical Information
Document Number: Q78557
Pubication Date: 17-SEP-1994
Product Name: Microsoft Disk Operating System
Product Version: 5.00 5.00a 6.00 6.20 6.21 6.22
Operating System: MS-DOS

If EMM386.EXE locks up the computer, follow these troubleshooting steps:

If the DEVICE=EMM386.EXE in the CONFIG.SYS file contains the HIGHSCAN parameter, remove HIGHSCAN from the command, save the CONFIG.SYS file, and restart your computer. (HIGHSCAN cannot be used on some computers.)

Start EMM386.EXE with the exclude option.

EMM386.EXE may have incorrectly identified an area being used by the system as a "hole" (a region that can be used as a UMB or an EMS page frame). As a result, EMM386.EXE overwrites a portion of memory being used by a hardware adapter in your computer.

By excluding addresses in the range A000-EFFF, EMM386.EXE does not use any part of the excluded region for a UMB or EMS page frame. Identifying the correct region(s) to exclude requires experimentation. Start by excluding a large region and then reducing the size of the region. For example:

```
DEVICE=EMM386.EXE NOEMS
X=A000-EFFF
```

```
DEVICE=EMM386.EXE NOEMS
X=C000-DFFF
```

```
DEVICE=EMM386.EXE NOEMS
X=C800-CFFF
```

You can use multiple exclusions on the EMM386.EXE device line. For example:

DEVICE=EMM386.EXE NOEMS X=C000-C7FF X=E000-EFFF

Refer to the documentation included with your hardware device(s) for information on the upper memory addresses the adapter card uses. Examples of devices which use upper memory addresses include network cards, SCSI adapters, video cards, and scanners.

How EMM386 and Windows 3.1 Relate (Include/Exclude Precedence)

Copyright Microsoft 1994
Title: How EMM386 and Windows 3.1 Relate
Document Number: Q82711
Pubication Date: 13-OCT-1994
Product Name: Microsoft Windows 3.x Retail Product
Product Version: 3.10 3.11
Operating System: WINDOWS

An X= parameter added to the EMM386.EXE device line is directly adopted by Windows. Any EMMInclude= or EMMExclude= entry is overridden by any X= parameter on the EMM386.EXE device line in the CONFIG.SYS file.

For example, the following line in your CONFIG.SYS takes precedence over any conflicting entry:

```
device=c:\windows\EMM386.EXE X=C000-CFFF noems
```

When Windows executes, it takes this information before it checks the SYSTEM.INI file. Even if the SYSTEM.INI file contains a contradictory line, for example:

```
EMMInclude=C000-CFFF
```

Windows keeps the EMM386.EXE parameter.

How to Troubleshoot Hardware Conflicts

Copyright Microsoft 1994
Title: How to Troubleshoot Hardware Conflicts

Document Number: Q90245
Pubication Date: 11-OCT-1994
Product Name: Microsoft Windows 3.x Retail Product
Product Version: 3.10
Operating System: WINDOWS

Hardware conflicts are difficult to diagnose because they manifest themselves in different ways depending on the type of network card and what they are conflicting with (disk driver, sound board, and so on). Typical hardware symptoms include:

Failure of the network card device driver to load; Error 5733 when the NET START command is used, or the network card fails to bind.

Inconsistent network behavior; the system stops responding (hangs) sometimes, or errors occur copying large files over the network.

Error 53: trying to connect to servers. (You cannot see anything on the network.)

If you suspect some type of conflict, try one of the following:

- Switch the network card to a different interrupt.
- Check for memory conflicts between the network card and other devices in the system. Some network cards use an of upper memory. If the area of memory, the card used is configurable. See the documentation for the network card.

You can determine which parts of upper memory are being used by other devices by doing the following:

Remove the network drivers and EMM386.EXE from CONFIG.SYS file.

Restart the computer.

Run Microsoft Diagnostics (MSD.EXE).

Look at the memory map. It should show any upper memory being used. If the net card is trying to use the same area as another device, reconfigure the network card to use a different memory area.

- Make sure the memory range for the network card (if it uses one) is excluded with EMM386.EXE if it is being used, or in the [386Enh] section of SYSTEM.INI file if EMM386.EXE in not being used.

4.3.3 Host PC Internal Bus Speed Conflicts

Strictly speaking, DSP/analog hardware is not expected to operate in a host PC with internal bus speed greater than 8.33 MHz, which was the original IBM-AT standard. However, over time, a situation has evolved where many boards of

different types can operate at higher bus speeds to provide better performance. And so, computer manufacturers have offered BIOS firmware options or motherboard switches which allow different internal bus speeds to be selected. Typically, one can select an internal bus speed that is the result of dividing the master PC clock speed by some integer. For example, on a 66 MHz 486 machine, options may be available for 13.2, 11, 9.43, and 8.25 MHz bus speeds. Often times, a machine has been configured by the manufacturer or system integrator to utilize hard disk controller, video card, etc. which can accept a higher speed, in order to provide better performance when delivered to the customer. Unfortunately, not all DSP/analog boards can tolerate increases above the 8.33 MHz mark. Cases where testing at Signalogic has shown this to be the case are indicated as highlighted notes in the individual hardware sections in chapter 3 of the Hypersignal-Macro Series Hardware Reference Guide. In such cases, you may need to run the BIOS configuration or setup program and look for an option to reduce the host PC internal bus speed.

Related Troubleshooting Topics

Troubleshooting Installation Problems

Troubleshooting Video and Display Problems

Troubleshooting DSP/Analog Hardware Problems

4.3.4 Data Acquisition Problems

This covers some typical problem areas that can occur during the course of using DSP/analog hardware. These are not installation or system conflict problems, which are covered in sections 4.3.1 through 4.3.3 above, I/O-Mapped Hardware Conflicts, Memory-Mapped Transfer Gotchas, and Host PC Internal Bus Speed Conflicts.

DC Offsets

DC offsets can be a minor but annoying problem. In some cases, they might be severe enough to subtract a significant amount of dynamic range from an A/D converter. Here are some things that might be done about it:

See if the DSP/analog hardware in use has a programmable "coupling" option; if so, choose AC coupling. Sometimes the coupling option is on a per-channel basis, and sometimes on an overall basis. Most DSP/analog hardware is fixed with DC-coupled, and

cannot be programmed.

See if the DSP/analog hardware has trim-pot adjustments, which can be used to control the dc offset of input amplifiers before the signal reaches the A/D converters.

If the offset is causing an artificial noise floor reading in a Spectrum Analyzer instrument display or other Fourier analysis function, try using no windowing and no zerofill in the FFT. The presence of any dc offset will cause a window to contribute its frequency characteristics to an FFT response.

Experiment with subtracting the offset digitally when recording analog data to hard disk or performing stimulus & response measurements. For example, the Hypersignal-Macro series software packages have "Input/Output Offset" fields which can be entered to non-zero values for offset control. Remember,

though, in this case the offset is merely being compensated for, and not removed, from the analog I/O system. If a full dynamic range signal has already clipped due to an offset problem, attempting to remove the offset digitally will be futile.

Related Data Acquisition Problem Topics

DC Offsets

Continuous Disk Transfer Performance

Continuous Disk Transfer Performance

One of the most common areas of concern when using DSP/analog hardware is maximum transfer rates to and from hard disk. To achieve maximum performance, here are some pointers and suggestions:

A fast system. Nothing helps like raw talent. A fast processor (for example, a 66 MHz 486 or better), a fast hard disk, preferably with its controller card located on a VESA local bus or PCI bus, video controller card located on a VESA local bus or PCI bus, and an internal bus speed of at least 8.33 MHz,

and possibly higher, will make a big difference. Locating the hard disk controller and video controller cards on a local bus is important, because the DSP/analog hardware should then have exclusive use of the standard ISA bus during acquisition (99% of off-the-shelf DSP/analog hardware are ISA bus plug-in cards).

Some network adapter cards and other peripheral cards can actually slow down a host PC's internal bus (ISA bus) to rates as low as 4.77 MHz because of the way they respond to bus acknowledgment signals. If you suspect your DSP/analog hardware disk performance is too slow, you might try temporarily taking out all other peripheral boards in your system, and re-measuring.

Make sure your available hard disk space is not fragmented. Non-contiguous file space can cause delays in

continuous acquisition to disk, which could translate into gaps in the data. You can use an application like Norton Desktop to periodically run a disk-defragmentation utility, say once a week. This should ensure large chunks of free, contiguous disk space at any given time. Also, if the software being used allows it, you might try increasing the number of buffers being used; more buffers act as a greater "shock absorber" for delays caused by non-contiguous file space or bad-tracks.

Do not use disk space compression; e.g. DoubleSpace, Stacker, etc. These utilities take extra time to compress data before it is actually stored to disk. Normally, this time is not noticed because a big memory cache is enabled for the system's disk I/O, and data is written first to the cache, allowing programs to think the transfer is complete.

But, when long data acquisition operations take place, the cache quickly gets full, and data must then be stored to disk at the same rate as it is coming in. At this point, the performance degradation of a compression utility will be quite noticeable.

For some systems, continuous disk transfer performance is actually increased if 32-bit disk access is disabled. This option is located under Program Manager | Main | 386 Enhanced | Virtual Memory.

For some systems, continuous disk transfer performance is actually increased if the SMARTDRV.EXE utility is not loaded. This option is usually specified inside the system's AUTOEXEC.BAT file, and sometimes in its CONFIG.SYS file (in the form "device = C:\WINDOWS\SMARTDRV.EXE" or similar). The reason? Because long,

continuous disk acquisition operations at high sampling rates eventually fill up the internal memory cache used by SMARTDRV.EXE; afterwards, the advantages of caching are lost as data must leave the cache at same rate it enters. At this point, the overhead of cache processing does not contribute to disk performance, and may detract. Note, however, that disabling SMARTDRV.EXE will have a detrimental effect on the performance of your system overall, especially for word processing, database, communications, and other applications that make extensive random file accesses. If disabling SMARTDRV.EXE turns out to be necessary for the DSP/analog hardware in your system to achieve continuous acquisition at a certain required maximum sampling rate, you should wrap the disabling into a

macro program or DSP engine function, so that it can be disabled and re-enabled automatically.

You can use the "sliding window" FFT technique to check for discontinuous data in a recorded waveform file that you suspect as possibly having gaps. To do this, perform an FFT function (in Hypersignal-Macro series software, use FFT Generation, in DSPower-Block Diagram use the FFTEx function) with the following parameters:

Framesize of 250
FFT size of 512 (FFT order = 9)
Overlap of 50%
Hamming Window

This will cause the input waveform to be analyzed with a sliding window that will highlight sharp transitions such as likely to be associated with a discontinuity. Showing the results in the 2-D spectrograph (contour) display function; can do two things: 1) display the entire waveform at once, and 2) allow a "look down" into the frequency data from 0 to -70 dB or -80 dB. If there are gaps in the data they should show up as one or more brightly colored vertical lines, and possibly evenly and/or repetitively spaced. The vertical line occurs because the discontinuity is seen by the FFT analysis as wideband noise; the even and/or repetitive spacing--if present--may represent the internal buffer size used by the acquisition operation.

Related Data Acquisition Problem Topics

DC Offsets

Continuous Disk Transfer Performance